THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Traitement du signal

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Synthesis and expressive transformation of singing voice

Présentée par

Luc Ardaillon

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

soutenue le 21 novembre 2017

devant le jury composé de :

M. Axel ROEBEL	Directeur de thèse
M. Thierry DUTOIT	Rapporteur
M. Nathalie HENRICH	Rapporteur
M. Christophe D'ALESSANDRO	Examinateur
M. Olivier ADAM	Examinateur
M. Jordi BONADA	Examinateur

Abstract

State-of-the-art singing voice synthesis systems are already able to synthesize voices with a reasonable quality, allowing their use in musical productions. But much efforts are still necessary to obtain a quality similar to that of a real professional singer. This thesis aimed at conducting research on the synthesis and expressive transformations of the singing voice, towards the development of a high-quality synthesizer that can generate a natural and expressive singing voice automatically from a given score and lyrics.

Due to the important variability of the voice signal, both from the control and timbral point of views, this involves considering various aspects. Mainly 3 research directions can be identified: the methods for modelling the voice signal to automatically generate an intelligible and natural-sounding voice according to the given lyrics; the control of the synthesis to render an adequate interpretation of a given score while conveying some expressivity related to a specific singing style; the transformation of the voice signal to improve its naturalness and add expressivity by varying the timbre adequately according to the pitch, intensity and voice quality. This thesis provides some contributions in each of those 3 directions.

First, a fully-functional synthesis system has been developed, based on diphones concatenations, which we assume to be up to now the approach capable of providing the highest sound quality. The modular architecture of this system allows to integrate and compare different signal modeling approaches.
Then, the question of the control is addressed, encompassing the automatic generation of the f_0, intensity, and phonemes durations. A particular limit of state-of-the-art approaches is a lack of controls provided to the composer to shape the expression of the synthesized voice. To tackle this issue, an important contribution of this thesis has been the development of a new parametric f_0 model with intuitive controls. The modeling of specific singing styles has also been addressed by learning the expressive variations of the modeled control parameters on commercial recordings of famous singers to apply them to the synthesis of new scores.
Finally, some investigations on expressive timbre transformations have been conducted, for a future integration into our synthesizer. This mainly concerns methods related to intensity transformation, considering the effects of both the glottal source and vocal tract, and the modeling of vocal roughness.

Résumé

Les systèmes de synthèse de voix chantée actuels sont déjà capables de synthétiser des voix avec une qualité raisonnable, permettant une utilisation dans le cadre de productions musicales. Mais beaucoup d'efforts sont encore nécessaires afin d'obtenir une qualité comparable à celle d'un réel chanteur professionnel. Le but de cette thèse était de conduire des recherches sur la synthèse et transformation expressive de voix chantée, en vue de pouvoir développer un synthétiseur de haute qualité capable de générer automatiquement un chant naturel et expressif à partir d'une partition et d'un texte donnés.

Du fait de la grande variabilité du signal vocal, tant du point de vue de son contrôle que de son timbre, cela implique de considérer des aspects variés. 3 directions de recherches principales peuvent être identifiées: les méthodes de modélisation du signal afin de générer automatiquement une voix intelligible et naturelle à partir d'un texte donné; le contrôle de la synthèse, afin de produire une interprétation d'une partition donnée tout en transmettant une certaine expressivité liée à un style de chant spécifique; la transformation du signal vocal afin de le rendre plus naturel et plus expressif, en faisant varier le timbre en adéquation avec la hauteur, l'intensité et la qualité vocale. Cette thèse apporte diverses contributions dans chacune de ces 3 directions.

Tout d'abord, un système de synthèse complet a été développé, basé sur la concaténation de diphones, que nous supposons être jusqu'à aujourd'hui l'approche capable de produire les résultats de la plus haute qualité. L'architecture modulaire de ce système permet d'intégrer et de comparer différent modèles de signaux.
Ensuite, la question du contrôle est abordée, comprenant la génération automatique de la f_0, de l'intensité, et des durées des phonèmes. Une limite particulières des approches de l'état de l'art est le manque de contrôles fournis au compositeur pour modifier l'expression de la voix synthétisée. Afin de résoudre ce problème, une importante contribution de cette thèse a été le développement d'un nouveau modèle de f_0 paramétrique intégrant des contrôles intuitifs. La modélisation de styles de chant spécifiques a également été abordée par l'apprentissage des variations expressives des paramètres de contrôle modélisés à partir d'enregistrements commerciaux de chanteurs célèbres, afin de les appliquer à la synthèse de nouvelles partitions.
Enfin, des investigations sur diverses transformations expressives du timbre ont été conduites, en vue d'une future intégration dans notre synthétiseur. Cela concerne principalement des méthodes liées à la transformation de l'intensité, considérant les effets liés à la source glottique et au conduit vocal, et la modélisation de la raucité vocale.

Remerciements

En premier lieu, je souhaite ici remercier mon directeur de thèse Axel Roebel pour m'avoir offert l'opportunité de faire mes premiers pas dans le milieu de la recherche en travaillant sur un sujet aussi riche qu'est la synthèse et transformation de la voix chantée me permettant ainsi de relier mon parcours scientifique à ma passion pour le son et la musique, pour son accompagnement tout au long de la thèse, et pour le partage de ses connaissances. Un grand merci également à Gilles Degottex pour m'avoir soutenu et accompagné durant mes premières années, pour sa relecture attentive de mon premier article, et son aide providentielle pour la mise en place de tests d'écoute qui m'auront servis jusqu'au bout de la thèse. Merci bien évidemment à Céline Chabot-Canet qui m'a éclairé de sa science musicologique pour tenter de ressusciter Edith Piaf. Merci plus largement à tout les collègues du projet ChaNTeR avec qui j'ai pu collaborer: Christophe, Vincent, Olivier D., Marius, Bruno, Olivier P., Lionel, et tout particulièrement à Sam pour nos aventures Porquerolloises, Suèdoises et musicales extra-ChaNTeResques. Et merci aux 3 chanteurs (Marlène, Raphael et Eléonore) qui ont accepté de nous prêter leurs voix pour se les faire triturer dans tout les sens par nos algorithmes, et pour avoir gentiment subis nos scéances d'enregistrement en s'appliquant à répondre à nos requêtes les plus saugrenues.

Ensuite je souhaiterai remercier Jean-Julien Aucouturier pour m'avoir permis de terminer cette thèse dans d'excellentes conditions et de mettre mes compétences à contribution pour faire crier des voix en toute séreinité, ainsi que pour l'opportunité de poursuivre ce travail ensemble pendant encore quelques mois. Pour ça merci également à Marco Liuni dont je suis la trace depuis l'équipe analyse-synthèse vers l'équipe CREAM, et qui a permis de créer des ponts entre nos recherches respectives. Merci également à l'ensemble de l'équipe CREAM.

Evidemment un grand merci à tout mes amis et collègues passés et présents de l'équipe analyse-synthèse: Nicolas et Geoffroy pour nos nombreux échanges et les conseils prodigués; Ugo avec qui j'ai partagé quelques années depuis ATIAM jusqu'à sa fin de thèse, et qui m'a apporté pendant tout ce temps un brun de lumière depuis son bureau sous la verrière jusqu'au côté obscure des sous-sols ircamiens; David qui a aussi fortement contribué à m'apporter cette lumière reflétée du bout de son trombone lors des répètes de l'éphémère fanfare du midi ou le soir entre 1 bière et un verre de shlivo; Stefan pour ses élucubrations poétiques et philosophiques; Maxime qui m'a aidé le temps d'un stage à rendre la voix plus douce, plus forte, plus expressive; tout les copains doctorants arrivés en cours de route (Céline, Hugo, Alice, Damien, Tristan) qui ont pris le relai pour repeupler nos bureaux de leur présence chaleureuse et joviale et particulièrement pour le soutien en fin de rédaction; et tout les autres que je ne saurais malheureusement citer sans risquer d'en oublier . . . .

Une petite dédicace également aux amis de ma promo ATIAM, et particulièrement Hélène, Vincent, et Hélianthe (et encore Ugo) pour les quelques bières partagées pendant nos années doctorales communes entre ces murs, en leur souhaitant le meilleur pour la suite.

Je souhaite saluer ici aussi Boris Doval pour sa pédagogie, qui a su susciter mon intérêt pour la recherche sur la synthèse et transformation de la voix lors de

ses cours en ATIAM.

Merci à Arnaud Petit de nous avoir offert l'opportunité d'une première application artistique de nos recherches.

5 ans ont passés depuis mon entrée à l'IRCAM en ATIAM, et je souhaiterai encore ici remercier tout ceux que j'ai pu rencontrer tout au long de ces années et avec qui j'ai pu avoir des discussions passionantes autour des sciences, de la recherche, et de la musique, ou quoi que ce soit d'autre.

Merci à mes parents et ma famille de m'avoir permis de suivre la [voie] que j'ai choisi jusqu'ici. Merci aux copains de Lack'o lock pour leur soutien également ces derniers mois pendant la rédaction (et pour la cuisine, le ménage, tout ça, je vais me rattraper), à Zarhzä (R.I.P?) le petit chat pour sa courte mais chaleureuse présence, et puis les autres Zarhzä et Calamity Street pour m'avoir permis de m'aérer la tête pendant les répètes.

Je souhaite aussi adresser mes remerciements à tout ceux (amis, famille, collègues, chercheurs, stagiaires, doctorants, inconnus, . . .) qui ont pris le temps de répondre à mes nombreux tests d'écoute malgrès les différences parfois subtiles entre les sons.

Enfin, merci aux rapporteurs et examinateurs pour leur temps et leur investissement en acceptant notre invitation à faire parti de mon jury de thèse, en espérant que mon travail aura su vous intéresser.

Contents

List of Figures

List of Tables

List of Abbreviations

AR	**A**uto-**R**egressive (filter, model)
ARMA	**A**uto-**R**egressive and **M**oving **A**verage (filter, model)
ASR	**A**ttack-**S**ustain-**R**elease amplitude envelope
DAP	**D**iscrete **A**ll-**P**ole (AR spectral envelope estimation method)
DFT	**D**iscrete **F**ourier **T**ransform
EGG	**E**lectro**G**lotto**G**raphy
FIR	**F**inite **I**mpulse **R**esponse (filter)
HMM	**H**idden **M**arkov **M**odel
HNM	**H**armonic + **N**oise **M**odel
IIR	**I**nfinite **I**mpulse **R**esponse (filter)
ISiS	**I**rcam's **Si**nging **S**ynthesizer
LF	**L**iljentcrant-**F**ant (glottal source model)
LPC	**L**inear **P**redictive **C**oding (AR spectral envelope estimation method)
LS	**L**east **S**quare
MFA	**M**ulti-**F**rame **A**nalysis
MFCC	**M**el-**F**requency **C**epstral **C**oefficient
MSE	**M**ean **S**quared **E**rror
PaN	**P**ulse **a**nd **N**oise parametric synthesis engine
PSOLA	**P**itch **S**ynchronous **O**verlap-**A**dd
PSY	**P**arametric speech analysis, transformation and **SY**nthesis
RMS	**R**oot **M**ean **S**quare
SDIF	**S**ound **D**escription **I**nterchange **F**ormat
STFT	**S**hort **T**erm **F**ourier **T**ransform
STRAIGHT	**S**peech **T**ransformation and **R**epresentation using **A**daptive **I**nterpolation of wei**GHT**ed spectrum
SVLN	**S**eparation of the **V**ocal-tract with a **L**iljencrantd-Fant model + **N**oise (voice model)
SVS	**S**inging **V**oice **S**ynthesis
TE	**T**rue **E**nvelope (spectral envelope estimation method)
TTC	**T**ext **T**o **C**hant (synthesis system)
VTF	**V**ocal **T**ract **F**ilter
VTS	**V**ocal **T**ract **S**hape
VUF	**V**oiced/**U**nvoiced **F**requency
WBVPM	**W**ide-**B**and **V**oice **P**ulse **Modeling**

Chapter 1

Introduction

1.1 Context and scope of this thesis

1.1.1 Synthesis and transformation of the singing voice

The synthesis of the singing voice is about as old as the synthesis of speech as a research subject and goes back to the 60's. It consists in generating a sound waveform (a sound file or a real-time sound stream) of a human voice, singing either according to a given score and lyrics, or controlled in real-time using an appropriate user interface.
When dealing with the transformation or synthesis of a singing voice, many different aspects have to be considered, from the modeling of the voice signal encompassing a very wide range of possible timbres and articulations, to the musical expressivity and its control for various possible singing styles.
The two main goals of conducting research on singing voice synthesis and transformation are:

- To get a better knowledge and understanding of the singing voice, from its technical, acoustical and interpretive aspects
- To bring new possibilities to the field of artistic creation (music, movies, ...)

Nowadays, speech synthesis has already reached a very satisfying quality for many applications, and singing voice is not far behind. But it is nevertheless still an active research field, as more efforts are necessary to reach a quality similar to that of a real professional singer, and continue to explore the very wide range of possible timbres and expressions of the human voice.

1.1.2 The ChaNTeR project

This thesis has been conducted at IRCAM, in the Analysis/Synthesis team, in the context of the *ChaNTeR* project [1]. This project was a collaboration between IRCAM [2], the Limsi [3], and the companies Acapela Group [4] and Dualo [5], whose main goal was to build high-quality singing voice synthesis systems with both real-time and offline control possibilities.
The role of IRCAM in this project (and especially in the scope of this thesis) was to conduct the research for building an offline synthesis system, which should be controlled from a text and a music score (e.g. in the midi or musicXML format).

[1] ANR project "ChaNTeR" : ANR-13-CORD-011
[2] https://www.ircam.fr/
[3] https://www.limsi.fr/en/
[4] http://www.acapela-group.com/
[5] https://dualo.org/

This kind of system are usually denoted as "Text-To-Singing", or "Text-To-Chant" (TTC) systems. Another goal was to be able to apply various singing styles to the synthesis.

1.1.3 Voice-related research at IRCAM

Voice has been a strong thematic in the Analysis/Synthesis team of IRCAM from its creation, and this thesis has benefited from a strong background on the analysis, modeling, synthesis and transformation of the voice, with applications in the field of artistic creation and beyond [Rod09] .
History of voice-related research at IRCAM goes back to the work conducted in the 80's by Xavier Rodet on the synthesis of the singing voice, with the *Chant* system [RPB84] .
Since then, lots of work have been conducted in the team in various voice-related areas. A few examples of the investigated areas are text-to-speech synthesis [Obi11; LDR10], expressive voice transformations [Bel09], gender and age transformation[6][7], speech segmentation [Lan+08], voice conversion [Hub15], source-filter separation [Deg10], or voice casting [ORB14].
The work carried out in this thesis therefore continues those effort towards a better understanding and modelization of voice signals.

1.1.4 Objectives and scope of this thesis

The main objective of this thesis was to develop a high-quality singing voice synthesis system that can, from a score and a text, automatically generate a sound file of a singing voice which sounds as natural and expressive as possible, as will be defined below.
To achieve this goal, many steps have to be performed. This possibly involves: some language and symbolic processing to phonetize the input text into a sequence of phonemes that is coherent with the notes of the score; the generation of all necessary control (prosodic) parameters, like the pitch or intensity curves and the phonemes durations; and the signal modeling and transformation part (depending on the chosen method) that generates the signal with the desired timbre and expressions. Figure 1.1 shows the basic building blocks of such a system, namely the system's inputs, the control module that generates the parameters required for the synthesis, and the synthesis module that generates the sound.

For some approaches, like concatenative synthesis, which is based on the use of samples recorded from a real singer, many transformations have to be performed (e.g. to change the pitch, duration, intensity, or voice quality of the recorded samples). For other methods, like physical modeling or formant synthesis, those features are inherent to the signal modeling. Singing style is also an important aspect to be considered for synthesis, that has implications in both the control and signal modeling parts.

Each of those steps should be thoroughly studied, in order to produce a high-quality synthesis with appropriate controls for the user for a large range of voice timbres and singing styles, and would require way more than a single thesis to fully achieve this goal.
This thesis thus could have focused only on a single, or a restricted subset, of those

[6] http://www.fluxhome.com/products/plug_ins/ircam_trax-v3 ,
[7] http://anasynth.ircam.fr/home/english/software/supervp-trax-en
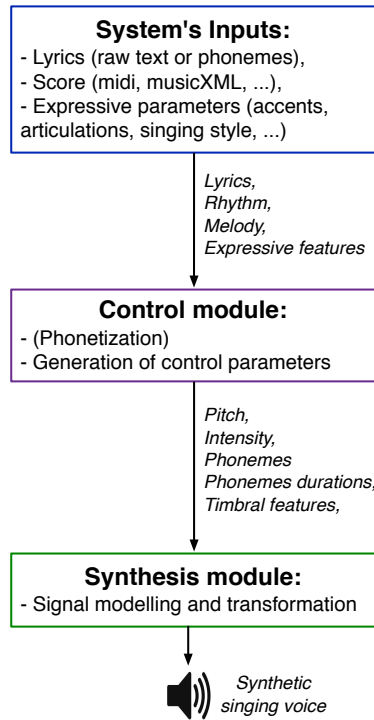
FIGURE 1.1: Basic building blocks of a TTC system

different issues. But on the other hand, some interdependencies exist between the different aspects, that can't be ignored. For instance, it would be hard to correctly evaluate the control of the synthesis if the signal transformations that are driven by the given control inputs have a bad quality, because the result of the synthesis would be too much degraded even though the generated control parameters may be appropriate. An example would be the quality of the transposition, that depends on the applied transposition factors, and thus on the generated fundamental frequency (f_0) curve. This raises the additional problem of evaluating the output of our synthesizer, which would also have to be considered. Also, the voice timbre, like that of any instrument, is not static and changes according to the pitch or intensity, for instance. The control and the modeling of the voice thus can't be considered as completely independent domains, and it thereby seemed relevant to first consider the synthesis system in its globality to take those considerations into account.
The work presented in this thesis thus relates to most of the necessary steps towards building an high-quality, fully-functional TTC system. This mainly concerns the signal modeling, transformation, and control parts.

However, thanks to previous research, some of the necessary tools and algorithms were already available with a rather good quality to perform part of this work, and could be used as a starting point, at the beginning of this thesis. This especially concerns the signal modeling and transformation parts, which can make use of tools like superVP[8]. Some preliminary work on concatenative singing voice synthesis had also been already done during an internship, before the start of this thesis [Ard13] .
Based on this available background, a TTC system based on diphones concatenation, called ISiS (for Ircam Singing Synthesizer), has first been built. Each part of

[8] http://anasynth.ircam.fr/home/english/software/supervp

the system has then been incrementally improved both in quality, controllability, and flexibility, according to the results of researches on those different aspects. Some of the research results that will be presented in this manuscript have not been integrated in the system yet, but offer interesting perspectives from future improvements of the system.

Another objective of the presented work was to be able to synthesize various singing styles, which relates to both the control of the synthesis, and some specific timbre transformations, as different singing styles, like lyrical or rock, require very different voice qualities.

To summarize, in addition to the development of a state-of-the-art concatenative synthesis system, this thesis mainly addresses 3 issues:

- modeling the appropriate control parameters from a score and lyrics
- building algorithms for natural-sounding timbre transformations
- modeling various singing styles, based on the available control parameters and timbre transformations

1.2 The singing voice as an object of study

1.2.1 Specificities of the singing voice: singing versus speech

Emanating from the same physical system, speech and singing have a lot in common. Logically, many techniques first developed for synthesizing speech have thus been adapted quite successfully for singing voice (unit selection, HMM-based synthesis, ...). But while everyone knows how to speak, not everyone knows how to sing well. And similarly, synthesizing good quality singing voices is not as easy as just constraining a text-to-speech system to follow the melody given from a score. Some specificities that differentiate singing from speech have to be considered.

Apart from specific contexts (e.g. a political speech, or an actors' performance), speaking is a relatively spontaneous process, and one mainly need to think about the words to be pronounced in order to express our thoughts, not about the way we pronounce those words. We don't think about how we raise or lower our voice while speaking, or which syllable should be accentuated. This process is referred as "prosody" and is the result of some implicit knowledge of a particular language (at least in the case of ones mother tongue). We don't think neither about how much pressure should be pushed from our lungs or how to place our larynx. Having learned to speak from our early childhood, we are all natural experts at it. But singing is a much less natural process. Lots of technique has to be learned to reach a good level, which requires years of training.

In speech, the main focus is usually on the message to be delivered, and thus on the intelligibility of the text that is pronounced. To some extent, some emotions or intentions are also conveyed, for which the words only are not enough. Then, the prosody helps to add some informations by shaping the pitch and dynamic of the voice to encode those "hidden" messages. But this implicitly happens in a rather well defined way, which doesn't result from any aesthetic choice of the

speaker. For instance, it is a convention that the pitch should always be raised at the end of a question. These various aspects of vocal communication have been already thoroughly studied from various angles [Fón83].
At the contrary, if the intelligibility is also important in singing, the main focus is usually on the aesthetic qualities of the voice, which mainly depends on its timbre and the various intonations produced, and this requires much more expertise and control from the singer than for speaking.

One big difference between speech and singing is that singing is usually constrained by a score that imposes, at least, the melody and rhythm of the song. In a way, this might thus seem easier to deal with, as this provides more informations on what the pitch should be, or where the phonemes should start, for instance, whereas in text-to-speech systems this has to be determined only from the linguistic content. But even with such informations, the expressive possibilities of the singing voice remain limitless, and result from artistic choices and cultural influences that can hardly be transcribed with a score and lyrics only.

Furthermore, one additional difficulty is that the range of possible pitch, intensity, speed, and voice quality is much wider in singing than in speech. In music, the voice can be for instance very high-pitched, very fast or very slow, very loud and intense, or very gentle, sweet or rough, etc... And an ideal TTC system should be able to cover all those possibilities. Modelling such variations of the voice thus requires new approaches to model and transform signals properly, that are not needed in speech synthesis, where the range of prosodic variations is much more restricted.

1.2.2 Diversity of vocal production in singing

Music, and especially singing, is a universal form of expression, but presents very diverse characteristics across countries, cultures, music genres, and social contexts. Many vocal styles exist that use the vocal apparatus in very different ways, exploring various possible timbral and expressive sub-spaces of the human voice.
We present here a few examples of those various singing techniques used across the world, and their specificities:

- **Soprano operatic singing**: singers can sing at very high pitches using the falsetto voice (specific laryngeal mechanism), and shape their vocal tract in order to tune their formants and thus maximize the homogeneity of the voice timbre [Gar+10; Ber+14] ;

- **Mongolian throat singing**: in order to reach a very low pitch range, singers make use of some specific vibratory mechanisms, with ventricular folds vibrating at half the frequency of the vocal folds, thus creating a period doubling phenomena [Lin+01] ;

- **Metal**: metal singers use very specific techniques in order to produce extremely rough timbres, also involving some supra-glottic structures as vibratory sources, in addition to their vocal folds [Nie08].

- **Belting**: belting is a non-classical singing technique used for instance by pop or musicals singers. It is characterized by a loud and bright sound and makes use of some resonance strategies that enhance higher harmonics [SM93; BS00].

- **And many more**: **croatian folk singing** [BK06], **indian classical ragga** [ABS09], **XXth century french variety** [Cha13], **african pygmy yodel** [Fri71], ...

With such diversity, it would be very ambitious to try modeling all those possibilities from the ground up in a unified framework. We thus have to fix some limits to our research, and choose one of direction to start with. Then, if the signal and control models used are flexible enough to model more diverse vocal productions, the synthesis system can be progressively extended to target more singing styles. In this thesis, we thus primarily focus on Western-European types of singing, such as lyrical and pop/variety singing styles. Especially, as part of the *ChaNTeR* project, we first target French singing and will thus try to imitate the singing styles of some famous French variety singers.

1.3 Why synthesizing singing voice?

There are multiple reasons for which one may be interested in synthesizing singing voices. A first one would be for purely scientific interest, in order to gain more knowledge about the way the voice is produced and perceived. In that case, some hypothesis about voice production or perception can be verified using synthesis, as we can assume that if we manage to synthesize a convincing singing voice, this means that we somehow understood the underlying process of producing such sounds for a human. This approach is usually referred to as "analysis by synthesis". Approaches based on physical modeling, or other ones based on signal models of voice production, are especially suitable for this purpose. But this is less true for some more recent approaches based on deep learning techniques, where only the sound is modeled and not the production process. Different techniques might thus be used for different purposes.

A second reason for synthesizing singing would be to discard the need of recording a real singer to interpret a score, when a singer with the right musical abilities or desired voice timbre is not available, or if the production environment is not adapted for recording in good conditions. This is particularly interesting for amatory music production in home studio conditions. Synthesis techniques are already in use in this context for many musical instruments, but not much for the singing voice yet, although some software like Vocaloid[9] are already used for this purpose. Singing voice synthesis thus enables the possibility to include vocal tracks in compositions without any other needs than a computer with the right software, and can be done anywhere with a laptop, thus offering both an economic and a mobile solution. A particular advantage is that the composer could, with a flexible enough software, parametrize it to choose a particular type of voice timbre and expression that matches his musical idea.

Synthesis could also help a composer to have a first rendering of his composition, to hear the result including the vocal part, as can already be done in most music score editors for many instruments, even though he/she prefers recording a real singer afterwards for a final version. But using a synthetic voice can also be an aesthetic choice, even though it still sounds a bit artificial. This is indeed probably what made the great commercial success of Yamaha's Vocaloid software, as people

[9] https://www.vocaloid.com/en

seem to like the recognizable sounding of the generated voices [Ken12]. The opera "The end", from keiichiro Shibuya, was created in 2013 based on the vocaloid software [10]. With this idea, the results of this thesis have also been applied to generate a synthetic voice for an opera of the composer Arnaud Petit, to be created in October 2017[11].

Another advantage of using synthesis is that the composer could have a precise control over the voice timbre and expression. He could for instance precisely tune the vibrato, the transitions between notes, the intensity variations, or the application of some timbre effects like growl at specific locations.

Some results from research conducted in the framework of singing voice synthesis (e.g. expression control and timbre transformations) could also be used in other contexts, as for instance for improving a real sung performance, as proposed in [Umb15].

Real-time synthesis is a particular case, where the voice articulation, expression, melody, rhythm, and timbre are controlled in live by means of a dedicated human-computer interface. The applications for this kind of systems would rather belong to the field of live music and performing arts.

A possible future application may also be to integrate the voice timbre and singing style learned from recordings of a famous deceased singer into a SVS system, and have this singer sing new posthumous songs (which might also be subject to more ethical questions), as has already been done for image using holograms of deceased singers (e.g. with Michael Jackson).

Finally, a particular interest of using voice synthesis is also the possibility to go beyond the limits of real human voices. It would thus be possible to have a computer-generated voice sing precisely some notes sequences with complex rhythms or intervals that would be very hard or impossible to sing for a real singer. One might also want to extend the ambitus of a singer, interpolate between different voices and singing styles as was done for the movie *"Farinelli"* [DGR95], or even create very specific voice timbres that don't sound human any more, but still present some characteristics of a singing voice.
Some other applications are also probably still to be found in the field of entertainment industry (cinema, video games, mobile phones apps, ...).

Nevertheless, our main interest, in the framework of this thesis, is to be able to generate voices that sound as expressive and natural as possible, with a variety of possible singing styles, and mainly for artistic purposes.

1.4 "Naturalness" and "expressivity": definitions

Before going further into the details of this work, and in order to explicit the goal of our research, it is necessary to define what is meant by the terms "naturalness" and "expressivity" in this thesis.

[10] https://www.youtube.com/watch?v=Ey8oj8S-j3U
[11] http://www.lefresnoy.net/panorama18/artwork/710/id/arnaud-petit

We call "naturalness" the property of a synthetic voice that could be thought as being a real recording of a human voice, as defined in [Rod02], without considering the aesthetic or artistic values of this voice. According to this definition, any vocal sound produced by a real person sounds "natural", even though this person is not a good singer.

The term "expressivity" designates the propensity of a synthesized voice to convey some musical intentions or emotions resulting from artistic choices and possibly related to a specific singing style, which makes the voice more musically interesting. The "expressivity" is basically what would make the difference between an average amateur singer and a professional one. As reported in [Umb+15], it might also be defined as *"the strategies and changes which are not marked in a score but which performers apply to the music."* (in [KM12]), or *the added value of a performance [which] is part of the reason that music is interesting to listen to and sounds alive* (in [Can+04]).

Both naturalness and expressivity are related to the timbral and prosodic features of the voice. But expressivity is more related to how those 2 key aspects vary in time and from one production to another, while naturalness is more related to how close to the physical reality of the voice production mechanism the used models are.

1.5 Main challenges in singing voice synthesis

As recalled earlier, the voice is probably the most complex and versatile of all acoustic musical instruments. Much research has already been devoted to the understanding of the mechanisms involved in singing voice production, and to its synthesis. But due to the non-static geometry of the voice organ, lots of characteristics and parameters are thus involved in all aspects of voice production, and more efforts are still necessary to reach the quality that could be expected from a real professional musician. According to the current state of researches, 3 main challenges can be identified for achieving the goal of building a truly natural and expressive-sounding SVS:

- The first challenge is to be able to produce an homogeneous, natural, and coherent timbre over an important range of pitch values. Whether the signal is transformed from a recorded sample or obtained using a parametric voice model, this requires the knowledge of the vocal tract filter of the voice, which can be obtained through spectral envelope estimation [Mak75; EM91; VRR06; RR05b] and source-filter separation techniques [Deg10; FM03]. But one issue is that the vocal tract filter (VTF) can usually only be partially observed, as its value can only be estimated at the partials' frequencies, for voiced sounds. This is especially problematic for high-pitched female voices, as the envelope is sampled by the harmonics with a poor resolution. Another aspect to be considered is that, for a given vowel, the vocal-tract filter is also pitch-dependant, as singers move their articulators (jaw, lips, tongue, ...) to keep an homogeneous timbre, and better project their voice [TW09]. The voice source should also be properly modeled, as its properties might also change according to the pitch, due to the different vocal folds vibratory mechanisms [RH09; Ber+14].

- A second challenge to be addressed is to enhance the expressive potential of synthesized voices through timbre modeling and transformations. A first objective would be to accurately reproduce the timbre variations related to voice intensity and vocal effort [LD99b; Mol+14]. Like for pitch, this has implications on both the voice source and the VTF. As said before, some singing styles are characterized by different timbral characteristics and can make use of various expressive timbral effects. An example of such effects is the growl effect [Sak+04; BB13], used in pop, jazz, or rock for instance, where some roughness has to be introduced in the voice. Other typical timbral characteristics of the voice are breathiness, tenseness, etc...

- A third challenging aspect of singing voice synthesis is its control. The questions to be answered are then: How to automatically control from the score all the prosodic and timbral variations of the voice? And which degree of controllability is available to the user? This question also encompasses that of singing style modeling, which has not been very much investigated yet.

This thesis mainly addresses the second and third of those challenges.

1.6 Main contributions

As evoked above, singing voice synthesis faces lots of issues on a variety of subjects, from musicology and physiology to signal processing and machine learning. Obviously, all issues could not be addressed in a single thesis, and choices had to be made to focus on a few ones.

The main contributions of this thesis that will be developed in this manuscript are:

- A thorough review of the state-of-the-art methods involved in various aspects of singing voice analysis, synthesis, and transformation, encompassing both signal modeling and expression control.

- The development of a fully-functional concatenative Text-To-Singing synthesis system called *"ISiS"*, for **I**rcam **Si**nging **S**ynthesizer.

- The proposal of a new multi-layer parametric f_0 model based on the use of B-splines, with intuitive controls to reproduce expressive features specific to singing voice.

- A new approach to model singing styles, using a rich description of the musical contexts along with some machine learning approach to learn expressive features of singers from a few recordings.

- An algorithm for producing a "mouth opening" effect to be used for natural-sounding intensity transformations.

- New approaches for introducing roughness in the voice timbre, useful for applying typical expressive effects (like growl) used in certain singing styles.

1.7 Outline of the manuscript

The organization of the document is the following:

First, a state of the art of the existing techniques related to singing voice synthesis (mainly voice modeling and transformations, and expression control), that served as a starting point to this thesis, is established.

The following chapter is dedicated to concatenative synthesis. *ISiS*, the singing synthesizer developed in this thesis, will be presented in its principles and architecture. The databases used for the synthesis will be described along with the synthesis engines that have been integrated in the software, and some problems related to the concatenation process will be tackled.

The subject of the next chapter is the control of the synthesis. The question addressed in this chapter is: How to automatically generate all the synthesis control parameters from the score and lyrics?
Our research on modeling the phonemes duration, the f_0, and the intensity curves will be presented. A particular focus of this chapter will be on the new $f_0$model proposed.

Then, this problematic is extended to the purpose of singing style modeling. Based on the results of the previous chapter and some established musical contexts description, a machine learning approach will be presented, that aims at extracting the expressive features of some singers from recordings and apply them to the synthesis of new songs.

The next chapter is dedicated to expressive timbre transformations. First, some work on modeling timbre modifications related to vocal effort and mouth opening for producing realistic intensity variations will be presented. Then, new approaches to roughness modeling with be described.

Finally, the main contributions of the thesis will be summarized, some ideas for improvements and future research directions will be exposed, and the problematic of the evaluation of singing voice synthesis will be shortly discussed.

In order to illustrate the presented work, some sounds are attached to this document, that can be accessed from the following url: http://recherche.ircam.fr/anasyn/ardaillon/these/these.php

Chapter 2

State of the art in modelization and transformation of the singing voice

This chapter aims at presenting the basic concepts and main techniques related to (singing) voice production, analysis, modeling, transformation, and synthesis required to get a good overview of the state of the arts techniques used in the field of singing voice processing, and to understand the researches presented in this thesis. More detailed explanations will be given for the specific concepts and techniques that have been used or studied in this thesis. For some key concepts, more in-depth explanations may also be given when required in the sub-sequent chapters.

2.1 Physiology of voice production

In this section, we will first describe the voice production system to give a basic idea of the main components and properties of voice. As any wind instrument, the human vocal apparatus is composed of a vibrating source excited by an air flow coming from a blower, and a resonator. Figure 2.1 describes the anatomy of the human vocal apparatus with its main components. The lungs play the role of the blower, expelling the air up to the trachea. The vocal folds are 2 parallel bands of mucous membrane situated in the larynx and form the vibratory source. The resonator of the vocal apparatus, called the vocal tract, is composed of everything above the glottis, from the larynx up to the lips and nostrils. Thorough explanations about the voice production system are given in [Sun90].

The myo-elastic theory of Van den Berg [Van58] explains the vibratory mechanism of the vocal folds. When air is pushed out from the lungs with a certain pressure (called "subglottic pressure"), this air column meets the vocal folds, initially closed. Due to the air pressure, the 2 folds then move away from one another. Then, due to the Bernoulli effect and some elastic return force, the vocal folds tend to stick back together again. Under certain physical conditions (mainly the level of subglottic pressure and vocal folds' tension), this action repeats at regular intervals and the vocal folds start to exhibit an auto-oscillation behaviour. The air flow, being modulated by the glottal opening, is then pulsed at a certain frequency in the vocal tract, thus creating an acoustic wave. The frequency of this sound is mainly dependant on the vocal folds' length and tension that are controlled by some dedicated muscles, and the intensity on the subglottic pressure. Those sounds produced by means of the vibration of the vocal folds are called "voiced sounds".
Due to viscosity and various constrictions of the vocal tract, part of the air flow
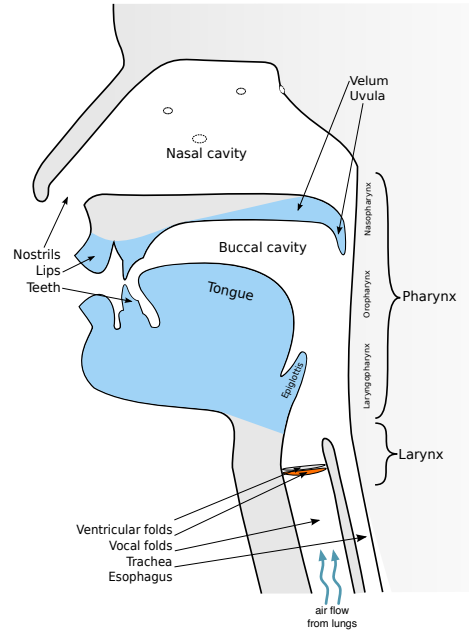
FIGURE 2.1: Description of the vocal production system (adapted from [Deg10], with the permission of the author)

also exhibits a turbulent behaviour, acting as a noise source. More specifically, these turbulences can be created either at the glottis level (termed aspiration noise), where it is thus modulated by the glottal opening, or at the tongue and teeth levels (termed frication noise). Sounds that are only composed of this stochastic noise component (e.g. fricative consonants, or whispered voice) are called "unvoiced sounds".
Other organs not used in usual phonation might sometimes also act as secondary vibratory sources, in addition to the vocal folds. This is the case of the ventricular folds (sometimes called "false vocal folds"), placed just above the vocal folds in the larynx, as shown in figure 2.1, which are implied in the production of rough voices [Bai09].

Then, the vocal tract acts as a natural filter that shapes the timbre of the sound produced by the voice source (both voiced and unvoiced). The shape of the vocal tract is determined by the position of all the articulators (higlighted in blue in figure 2.1). The main articulators used by singers to shape the sound are: the tongue, the jaw, the velum, and the lips. To a lesser extent, some muscles in the pharynx can also be used to apply some constrictions at different places. The vocal tract can then be modeled as a simple tube with a varying diameter determined by the position of those articulators. Depending on this shape, the vocal tract resonates at some particular eigen frequencies. These resonances of the vocal tract are called the "formants". Figure 2.2 shows the relation between the articulators' positions and the vowel produced, and figure 2.3 also shows the relation between those vowels and the two 1st formants' frequencies F_1 and F_2.

Although formants may be affected by all elements of the vocal tract, note that some articulators have a higher effect on certain formants. As explained in [Sun90], the first two formants are related to the produced vowel, the first formant being primarily related to the jaw opening and the second formant to the position
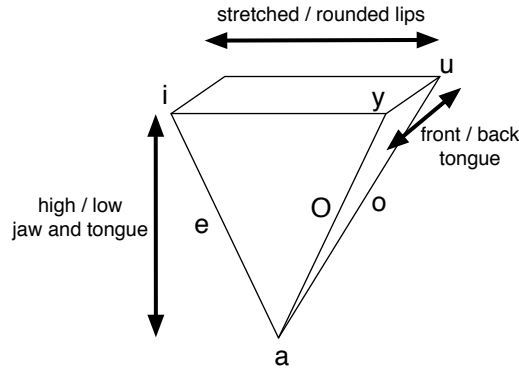
FIGURE 2.2: Relation between articulators positions and vowels. (adapted from [Fux12], with SAMPA phonetic notation)
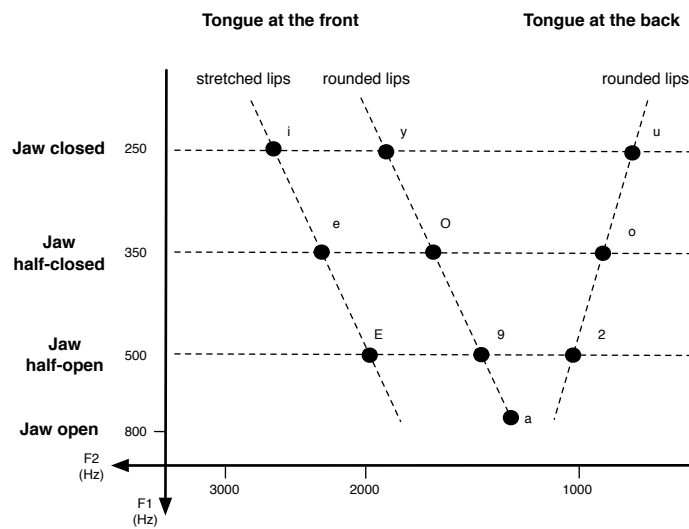
FIGURE 2.3: Vocalic triangle: Relation between formants positions, vowels, and articulators' positions. (adapted from [Fux12], with SAMPA phonetic notation)

of the tongue. The next three formants are more related to timbre and voice identity, the third formant being particularly influenced by the position of the of the tip of the tongue and the fourth one by the dimensions of the larynx.

In the case of nasal sounds (e.g. /a~/, /o~/, /e~/ in SAMPA phonetic notation, presented in annexe A), the velum is open, and the air flows both through the oral and the nasal cavities, and the vocal tract thus becomes composed of 2 parallel tubes. This configuration introduces some additional anti-resonances that are specific to those nasal sounds.

Finally, the sound is radiated at the lips (and nostrils for nasal sounds) level.

2.2 The source-filter modelization of voice

From the signal point of view, the voice mechanism is often modeled using the so-called source-filter model, which is an approximation of the behaviour of the vocal apparatus, seen as a linear system. Lots of algorithms and techniques used in the

field of voice processing are based on this rather simple model, which will also be used in the scope of this thesis. This section thus aims at giving an overview of this model, for a good understanding of the following sections and chapters.

2.2.1 Overview

In this representation, a generated source signal is simply shaped by a filter representing the resonances (and anti-resonances) of the vocal tract, assuming that this view is close enough to the reality from a perceptual point of view. The source and filtering parts are thus considered as independent components, discarding the effects of non-linear behaviours that may occur due to the various interactions between the vocal folds and the supra-glottic structure (vocal tract, ventricular folds, ...) and the influence of the time-varying geometry of the glottis during each glottal cycle.
A general formulation of the source-filter model in the frequency domain is given in equation 2.1.

$$S(\omega) = G(\omega) \cdot C(\omega) \cdot L(\omega) \tag{2.1}$$

where $S(\omega)$ is the spectrum of the radiated voice signal, $G(\omega)$ the source spectrum, $C(\omega)$ the VTF transfer function, $L(\omega)$ represents the radiation at lips and nostrils, and ω is the angular frequency.
The spectral shape of the source component is mainly dependant on the pitch, vocal effort, and voice quality. In this equation, the source $G(\omega)$ represents both the deterministic (voiced) and stochastic (unvoiced) parts. A more refined formulation detailing the source's components is thus given by equation 2.2

$$S(\omega) = ((H^{f_0}(\omega) \cdot G_{vo}(\omega)) + N(\omega)) \cdot C(\omega) \cdot L(\omega) \tag{2.2}$$

where $G_{vo}(\omega)$ represents the spectrum of the deterministic part of the source due to the vibration of the vocal folds, $H^{f_0}(\omega)$ is an harmonic comb at fundamental frequency f_0 that represents the periodicity of the glottal vibration, and $N(\omega)$ represents the noisy part (encompassing both aspiration and frication types of noise).
The lips (and nostrils) radiation is modeled as a simple time derivative, as explained in [Deg10]: $L(\omega) = j \cdot \omega$.
More details about the source-filter model can be found in [Deg10] or [Hub15].

2.2.2 Glottal source modeling

The source-filter model has been proposed as a simplified representation useful to understand the physiological fundamental of the voice and to give a convenient framework for voice processing, where the different components can be manipulated separately in a perceptually relevant way. For this purpose, it is thus necessary to have signal representations of these components, starting with the source component.

In the most simple version of source-filter model implementations, as the one used in early HMM-based speech synthesis systems [Yos+01], the source is modelled in the time-domain as a simple impulse train at a certain frequency for the deterministic part $G_{vo}(\omega)$, and white noise for the stochastic part $\sigma(\omega)$, all the timbral characteristics that color the voice being grouped into the filter part $C(\omega)$. But in reality, some timbral features are not related to the vocal tract, but rather to the vocal folds vibratory characteristics, and would thus better be treated
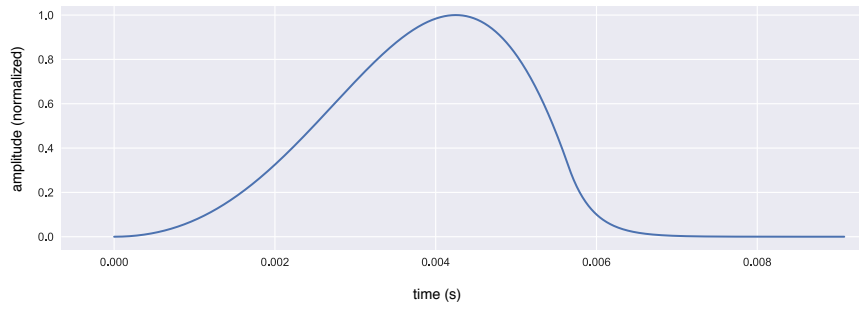
FIGURE 2.4: Typical shape for one period of the glottal flow of LF model

separately.
For better quality and flexibility, glottal source models have thus been proposed. Examples of such models are the CALM [DAH03] and the LF [FLL85] models, which is the one used in part of this thesis, and that we will thus present here.
Figure 2.4 shows a characteristic shape of the glottal flow in the time-domain for one period. As one can see, the period is composed of 2 phases: one during which the glottis is open, and one during which the glottis is closed and the flow is thus null.
As the presented source-filter model is composed of linear operators only, their
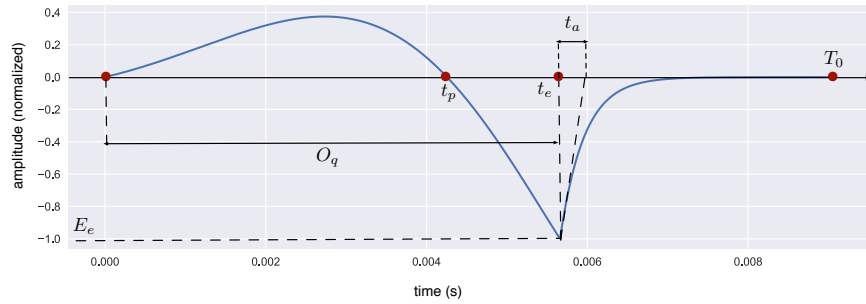
FIGURE 2.5: Derivative glottal flow with LF model parameters

order can be rearranged, and the radiation filter ($L(\omega)$) can thus be placed between the source and the vocal tract filter for convenience. For this reason, the LF model models the derivative of the glottal flow, rather than the glottal flow itself, in order to take into account the lips radiation.

The LF model is described in [FLL85], and figure 2.5 shows a typical shape for one period of the derivative glottal flow, as generated by the model, with its 4 parameters t_p, t_e, t_a, and E_e. t_p corresponds to the time of the maximal opening of the glottis, t_e to the time of the minimal value of the derivative during the closing phase, and the return phase t_a is equal to the duration between t_e and the point where the tangent at time t_e becomes equal to 0. Those values are relative to a normalized pulse period of 1 (with $T_0 = 1$, while the time axis in the figures is given in seconds). The fundamental frequency is thus required to scale them according to the real pulse period T_0. E_e is the minimal negative amplitude at t_e and sets the overall energy of the pulse.

TABLE 2.1: LF model parameters

parameter	range	description
T_0	$]0;+\infty[$	fundamental period
O_q	$]0;1[$	open quotient
α_m	$[0.5;1[$	open phase skewness
t_a	$[0;1-O_q[$	return phase
E_e	$[0;+\infty[$	pulse energy

An equivalent but more intuitive parametrization used to drive the LF model makes use of the parameters $O_q = \frac{t_e}{T_0}$, called the "open quotient", and $\alpha_m = \frac{t_p}{t_e}$, that characterizes the skewness of the open phase of the pulse [DAH06].
Table 2.1 summarizes the parameters of this LF model.

If this model can generate a good approximation of a real glottal source, using 3 parameters ([t_a,t_p,t_e] or [O_q,α_m,t_a]) to set the pulse shape, it might still not be the most convenient to handle for analysis and synthesis purposes. For this reason, Fant introduced an efficient one-dimensional parametrization of the pulse shape with the LF model using a single "meta-parameter" R_d, as described in [Fan95]. R_d is defined by equation 2.3:

$$R_d = \frac{U_0}{E_e} \cdot \frac{f_0}{110} \tag{2.3}$$

(where U_0 is the amplitude of the glottal flow, as shown in figure 2.4, and f_0 is the fundamental frequency ($= \frac{1}{T_0}$)), and equation 2.4, using the R parameters:

$$R_d = (\frac{1}{0.11})(0.5 + 1.2R_k)(\frac{R_k}{4R_g} + R_a) \tag{2.4}$$

where $R_a = \frac{t_a}{t_0}$, $R_g = \frac{T_0}{2t_p}$, and $R_k = \frac{(t_e - t_p)}{t_p}$.

Equation 2.4, obtained by means of a statistical regression of R_d values, as defined by equation 2.3, on a space of the co-varying underlying R shape parameters measured on various speakers, gives a means to compute the original LF model parameters from this single R_d parameter, as detailed in [Fan95]. Fant has shown that this parameter is the most effective one to describe voice qualities into a single value. This simple parametrization allows to describe voice qualities as a continuum of tense, modal and relaxed voice qualities. Lower R_d values correspond to more tense and higher R_d values to more relaxed voice quality, typical R_d values being found in the range $[0.3; 2.7]$ (and possibly higher at sentences boundaries) [Fan95].
From equation 2.3, one can see that for a constant R_d and F_0, parameter E_e is directly proportional to U_0 and thus relates to pulse energy, as written in table 2.1.

In the frequency domain, this glottal source is mainly characterized by its spectral tilt and a glottal formant, which corresponds to a resonance in the low-frequencies, in the vicinity of f_0. Figure 2.6 shows the derivative glottal pulse shape and corresponding spectrum for different values of R_d. Note that for low R_d values, the vocal folds remain closed for a large portion of the glottal cycle (low value

of open quotient O_q) and the voice has more high-frequency content (low spectral tilt), due to the short length of the glottal pulse and the rapid closure of the vocal folds, and inversely for high R_d values.
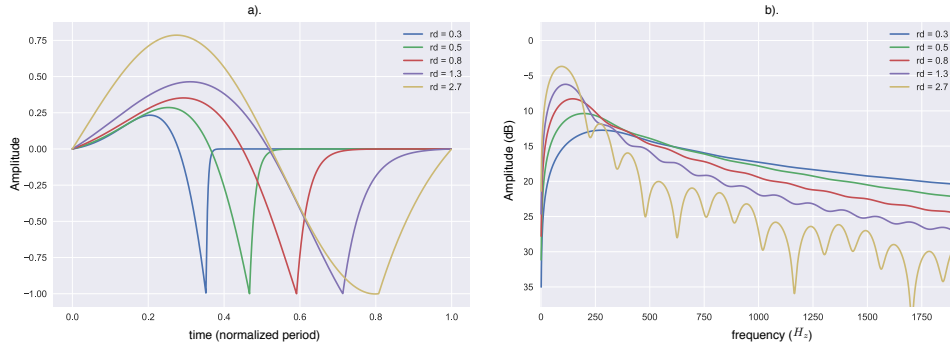
FIGURE 2.6: Examples of a). derivative glottal pulse shapes and b). corresponding spectrum, for various R_d values

This modelization of the deterministic source component can then be mixed with the stochastic component (e.g. modeled as filtered white or gaussian noise).

2.2.3 Spectral envelope estimation

We first presented the source component of the source-filter model. We will now present the complementary part of the model, i.e. the Vocal Tract Filter (VTF). As explained in section 2.2.1, the VTF shapes the source spectrum by applying the resonances (and anti-resonances) due to shape of the vocal tract. These resonances are called formants and greatly contribute to the voice timbre, characterizing the pronounced phonemes, or to some extent the vocal quality (related to expressivity or emotions), or the gender, age, and identity of a speaker. In order to synthesize or transform a voice appropriately, it can be necessary to estimate this VTF from real recorded voice signals. However, it is a difficult task to estimate directly the VTF, separated from the source contribution. It is also not necessary in many applications, for which it may be sufficient to assume a spectrally flat source (either white noise, an harmonic comb, or a combination of both). Rather than the VTF, we thus first seek to estimate the spectral envelope of the sound.

The spectral envelope can be defined as a smooth function passing through the prominent peaks of the voice spectrum, as illustrated in figure 2.7. In terms of the source-filter model, the so-defined spectral envelope thus represents both the contribution of the source spectrum and vocal tract filter as one unique function of frequency. Here we present the possible approaches for estimating this function from sound signals. This definition thus does not correspond to the $C(\omega)$ component from equation 2.2 which would be the VTF, but rather to the combination $G_vo(\omega) \cdot C(\omega) \cdot L(\omega)$. As we have seen in the previous section, we usually model the derivative of the glottal flow (corresponding to $G_{vo}(\omega) \cdot L(\omega)$ in the spectral domain) as the voice source. However, the problem of estimating and separating the source spectrum and VTF from the signal will be discussed in a next section.

Estimating the spectral envelope from a noisy signal (i.e. filtered white noise) is easy, as there is energy present at all frequencies. However, estimating it on harmonic signals, as it is mostly the case for voice, is much more difficult, because
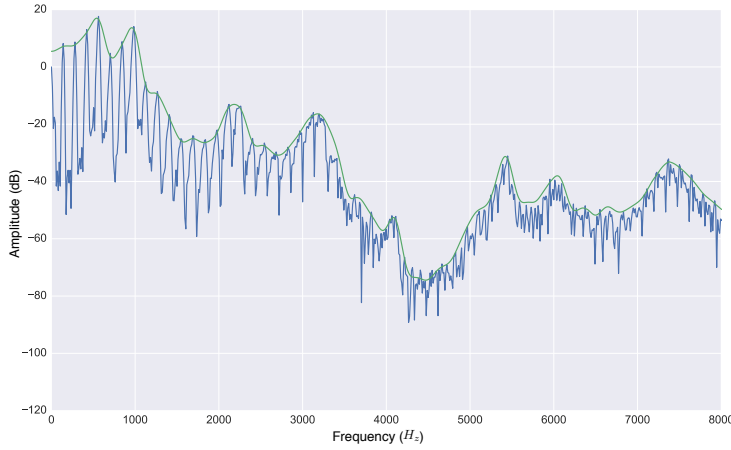
FIGURE 2.7: Spectral envelope (green) of a voice spectrum (blue)

the envelope is only sampled at the discrete harmonic frequencies (as shown in figure 2.7), which, depending on the f_0, may create aliasing. The main problem is thus to be able to reconstruct the continuous envelope from its known values at the sampled frequencies. This estimation is especially more difficult for high-pitched voices, as the sampling of the envelope becomes more sparse.

In [RVR07], the authors investigated the problem of estimating the spectral envelope for harmonic signals, comparing the main existing approaches, and giving some insight on the optimal parametrisation of the algorithms. Those approaches can be classified in 2 categories, that will be exposed below: cepstrum-based approach and methods based on all-pole models.

2.2.3.1 Cepstrum-based approaches

A first possible approach that is available for spectral envelope estimation is cepstral smoothing [Opp69]. The real cepstrum $C(l)$ is defined as the inverse Fourier transform of the log amplitude spectrum of a sound. If we define $X(k)$ to represent the K-point DFT of the signal's frame $x(n)$, we have:

$$C(l) = \sum_{k=0}^{K} log(|X(k)|)e^{i\frac{2\pi kl}{K}} \tag{2.5}$$

In the cepstral domain, the variable l is called the *quefrency*. Assuming $G(k)$ an harmonic comb and $H(k)$ the spectral envelope, we have :

$$X(k) = G(k)H(k) \tag{2.6}$$

Taking the log of the modulus, we have :

$$log(|X(k)|) = log(|G(k)|) + log(|H(k)|) \tag{2.7}$$

In the cepstral domain, the harmonic source component is found in the high quefrencies, while the spectral envelope is found in the low quefrencies. Thanks to the linearity of the fourier transform, the simple additive operator due to applying the logarithm thus allows to separate the 2 components. By setting all high quefrency

elements in the cepstrum above coefficient $P+1$ to 0, P being the cepstral order, we can thus isolate and retrieve the spectral envelope $H(k)$. As stated in [RVR07], the optimal order for an harmonic signal is

$$\hat{P} = \frac{F_s}{2f_0} \tag{2.8}$$

where F_s is the sampling rate and f_0 the fundamental frequency of the signal.

Unfortunately, the filtered cepstrum will create an envelope following the mean of the spectrum and not the contour of the spectral peaks, as would be desired. Based on the cepstrum, 2 main approaches allow to cope with this problem: the discrete cepstrum [GR90; CM96] and the True-Envelope (TE) [RR05a] algorithms.

- **Discrete cepstrum:**
  As stated in [CM96], the spectrum magnitude $|X(k)|$ is related to the real cepstrum coefficients c_i by:

  $$log(|X(k)|) = c_0 + 2\sum_{i=1}^{P} c_i cos(2\pi f i) \tag{2.9}$$

  The discrete cepstrum method consists in finding the parameters of such a cepstral model by means of a least square approximation using only the peaks of the signal amplitude spectrum, such that the obtained spectrum is close to what is considered a spectral envelope. But the problems of this method are that it requires a fundamental frequency analysis or another means to preselect the spectral peaks, that it is often ill-conditioned, and has computational complexity of $O(P^3)$.

- **True-envelope:**
  The True-Envelope algorithm [RR05a] performs an iterative cepstrum-based estimation, without requiring to select spectral peaks. Let $V_i(k)$ be the cepstral representation of the spectral envelope at iteration i, that is the Fourier transform of the filtered cepstrum. First we set $A_0(k) = log(|X(k)|)$ and $V_0(k) = -\infty$. Then the algorithm iteratively replaces the current target amplitude spectrum according to

  $$A_i(k) = max(A_{i-1}(k), V_{i-1}(k)) \tag{2.10}$$

  and iteratively applies cepstral filtering to the updated spectrum A_i. With this procedure, the valleys between the peaks of the target spectrum will be filled progressively by the cepstral filtering until all the peaks are covered. As stopping criterion of the procedure, a parameter δ is used that defines the maximum excess that a peak of the observed spectrum is allowed to have above the estimated spectral envelope (e.g. $\delta = 2dB$). The resulting estimation can be interpreted as the best band limited interpolation of the major spectral peaks. The true-envelope algorithm has been used in this thesis by means of its implementation in the SuperVP software [1].

[1] http://anasynth.ircam.fr/home/english/software/SuperVP

2.2.3.2 All-pole models

All-pole, or auto-regressive (AR), models are well-adapted for voice processing, as the resonances of the vocal tract (i.e. the formants) can be well represented by the poles of a corresponding all-pole IIR digital filter. Vocal sounds may also sometimes exhibit some anti-resonances, which would correspond to the zeros of a digital filter, for which an ARMA model would thus be better suited. However, zeros are only present in nasal sounds, or at some specific high-frequency locations (e.g. from 4 to $5kH_z$ due to the piriform sinus [DH97]), and not as perceptually relevant as formants for speech intelligibility and speaker identification. Anti-resonances are thus of secondary importance for voice, such that all-pole models usually give a good enough approximation of the spectral envelope for many processing tasks.

In auto-regressive models, a signal is explained as a linear combination of its past values and a white noise excitation source, as expressed by equation 2.11 :

$$x(n) = \sum_{k=1}^{P} a_k x(n-k) + u(n) \tag{2.11}$$

where x is the observed signal to be modelled, the a_k are the AR coefficients of the model, P is the model order, and u is the excitation signal.
In the Z domain, an all-pole model can be defined by its transfer function :

$$H(z) = \frac{G}{1 + \sum_{k=1}^{P} a_k z^{-k}} \tag{2.12}$$

where G is a fixed gain coefficient.
2 popular methods exists to estimate the AR coefficients a_k from a speech signal: linear prediction (LPC) [Mak75] and Discrete All-Pole modeling (DAP) [EM91].

- **LPC :**
  In order to find the best estimate of the coefficients a_k, we seek to minimize the quadratic error between the signal $x(n)$ and its model $\hat{x}(n)$:

  $$\sigma = (x(n) - \hat{x}(n))^2 \tag{2.13}$$

  It can be shown that

  $$\sigma = R(0) - \sum_{k=1}^{P} a_k R(k) \tag{2.14}$$

  where R is the auto-correlation sequence of the signal :

  $$R(i) = \sum_{n=-\infty}^{\infty} x(n)x(n+i) \tag{2.15}$$

  This result leads to a set of linear equations, known as the Yule-Walker equations, that can be written in matrix notation as :

  $$\begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(P) \end{bmatrix} = \begin{bmatrix} R(0) & R(-1) & \cdots & R(-P+1) \\ R(1) & R(0) & \cdots & R(-P+2) \\ \vdots & & \ddots & \vdots \\ R(P-1) & R(P-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{bmatrix} \tag{2.16}$$

From there, the coefficients can be estimated by inverting the matrix.
This approach is well suited to estimate the spectral envelope assuming a white noise excitation signal. However, for voiced speech sounds, the spectral envelope obtained from this method contains systematic errors as a consequence of the aliasing (in the auto-correlation domain) due to the subsampling of the spectral envelope by the harmonics, and will start to fit the spectral peaks, especially for high-pitched sounds. The DAP algorithm aims to solve this problem for a better estimation.

- **DAP :**
The basic idea exploited with the DAP algorithm [EM91] is to fit the all-pole model using only the finite set of spectral locations that are related to the harmonic positions of the fundamental frequency. The objective function used for this purpose in DAP is a discrete version of the Itakura-Saito measure, defined by

$$E_{IS} = \frac{1}{N}\sum_{m=1}^{N} \frac{P(\omega_m)}{\hat{P}(\omega_m)} - ln\frac{P(\omega_m)}{\hat{P}(\omega_m)} - 1 \tag{2.17}$$

where $P(\omega_m)$ is the given discrete spectrum defined at N frequency points ω_m and $\hat{P}(\omega_m)$ is the all-pole model spectrum evaluated at the same frequencies. It is then shown in [EM91] that minimizing this error leads to the following equations :

$$2\sum_{k=0}^{P} a_k \left[R(i-k) - \hat{R}(i-k)\right] = 0, \qquad 0 \leq i \leq P \tag{2.18}$$

where $R(i)$ is the autocorrelation corresponding to the given discrete spectrum and $\hat{R}(i)$ is the autocorrelation corresponding to the all-pole model sampled at the same discrete frequencies. These equations can then be solved using an iterative procedure to find the optimal filter coefficients a_k.
This approach requires to first select the spectral peaks to be used in the error measure and adds some computational complexity compared to LPC, but is assumed to give a better estimate of the spectral envelope. Note that for the continuous case of the presented error measure, the result would be the same as with the LPC.

All the algorithms presented here for spectral envelope estimation are implemented in the superVP software[2] [LR13]. The true-envelope algorithm leading to an accurate peak fitting has been mainly used throughout this thesis whenever spectral envelope estimation was required. However, having a parametric representation of the envelope such as an all-pole model is also useful for some processing tasks, and the DAP algorithm has thus also been used in this thesis for some timbre manipulation purpose based on poles modifications, as will be exposed in chapter 6. Figure 2.8 shows as an example the results of the spectral envelope estimation on an harmonic spectrum using the different methods presented. As can be seen on this figure, the true-envelope give the best approximation of the spectral envelope. The LPC and DAP approaches give very similar results, but completely miss the valley around 4500Hz, which is not represented by the all-pole model. Finally, the discrete cepstrum approach tends to oversmooth the envelope, and is thus not very

[2] http://anasynth.ircam.fr/home/english/software/supervp

accurate compared to the other methods.
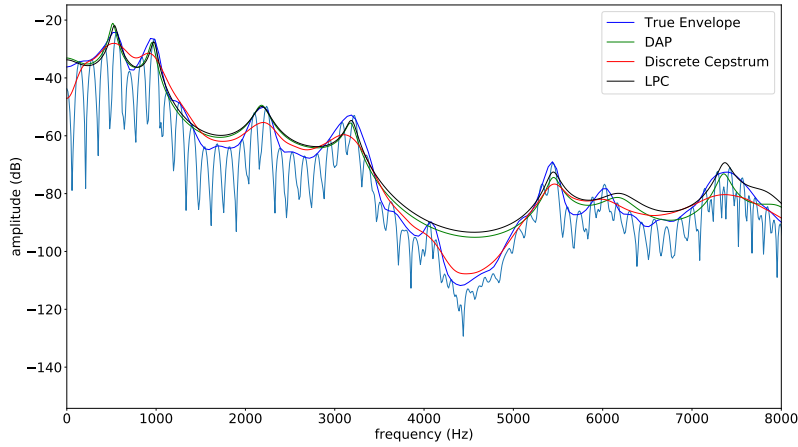
FIGURE 2.8: Spectral envelope of an harmonic signal estimated using various approaches

2.2.3.3 Multi-Frame Analysis (MFA)

As seen before, the accuracy and robustess of the spectral envelope estimation techniques is often limited by the sub-sampling of the real envelope by the partials of the signal. These estimation are ususally done on a single frame of the signal. However, assuming that the spectral envelope is relatively stable in time over several frames, and that f_0 is changing from one frame to another, the spectral envelope can be sampled more densely by combining several frames in the analysis. In this direction, several algorithms have been proposed for such multi-frame analysis of the spectral envelope [Deg15; SK03a].
In the framework of the ChaNTeR project, researches have also been conducted in this direction for better estimation of spectral envelope for morphing applications (e.g. pitch or intensity transformations), which gave encouraging results [DAR16a; DAR16b]. However, the assumption of having a stable envelope with a varying f_0 is not ensured, as the VTF may change for instance in synchronisation with the vibrato. This should thus be subject to further investigations to get more insight on the perceptual relevance of these considerations.

2.2.4 Source and filter separation

For expressive singing voice synthesis, advanced timbre manipulations are necessary. As mentioned previously, some timbral properties are mainly related to the source component, while others are only related to the VTF. Once the spectral envelope has been estimated, it may thus be useful to separate the source contribution from the VTF to be able to control those 2 components independently. For this purpose, it is necessary to first estimate the source shape parameters.
Many methods have been proposed to address this issue, among which pre-emphasis [MG76], Iterative Adaptive Inverse Filtering (IAIF) [Alk92], closed-phase analysis [PQR99], phase minization methods [Deg10], and others... However, this problem being only of secondary interest for the presented work and

for the sake of brevity, all those methods won't be detailed here. This question has been thoroughly addressed in [Deg10] which presents the different methods.

The algorithm that we used in this thesis, implemented into the superVP software, is based on the methods described in [DRR11a; Deg10] and [Hub15]. This algorithm aims at estimating the parameter R_d of the LF model, introduced in section 2.2.2. The main assumption behind the chosen approach is that the impulse response of the vocal tract is minimum phase, while the glottal source is a mixed-phase signal. The proposed R_d parameter estimation algorithm is thus based on objective functions for phase minimization, by fitting synthesized LF models, parametrized by different R_d values selected on a grid spanning the possible range, against the speech spectrum $S(\omega)$. The goal is to minimize the mean squared phase error between the signal's spectrum and its model when matching a synthesized glottal pulse against a strictly harmonic representation of voiced speech. In each frame analyzed, the R_d resulting in the lowest remaining phase error is selected.

However, the use of the 1-dimensional R_d parametrization restricts the synthesized LF model shapes to a subspace of its complete parameters space and thus results into an approximation, that may not perfectly fit the true glottal source shape of the analyzed signal.

With the proposed method, the optimal minimum-phase VTF is implicitly jointly estimated as the complement of the estimated source shape. It can thus be retrieved, by a simple spectral division of the signal's spectrum by the source's spectrum.

2.3 Singing voice synthesis techniques

Since the first works on singing voice synthesis in the 60's, much progress have been made, and the obtained quality is already good enough to be used in music productions, using popular commercial softwares such as Vocaloid [KO07]. In the articles [Coo98] from 1998 and [Rod02] from 2002, the authors give an overview of the state of the art techniques and applications at that time. But more progress have been made since then and new approaches have emerged. We review in this section the main methods used to synthesize singing voice, which can be divided into 4 broad categories: formant synthesis, physical modelling synthesis, concatenative synthesis, and statistical parametric synthesis (including HMM-based and neural-network-based approaches). A list of various research projects on singing voice synthesis using those approaches has been established in [Umb+15].

2.3.1 Formants synthesis

We call "formant synthesis" methods that are based on the source-filter model, usually using a source model generated as a mix of voiced and noise components, which is then filtered by a a set of filters simulating the vocal tract transfer function, with specific resonances corresponding to formants parametrized according to each phoneme. It is the oldest method for synthesizing voice, and many systems have used this approach, from the 60's to nowadays. Klatt described such a system for speech synthesis in [Kla80], where the filter can be modeled either as a cascade or a parallel connection of digital resonators, implemented in a hardware or a software environment. Other examples of formant-based singing synthesizers are

the *MUSSE* synthesizer [Ber96], and the *Cantor Digitalis*[3] [Feu13].

A specific type of synthesis that can be related to this category is the FOF synthesis (for "Formes d'Ondes Formantiques", which is French for formant wave functions), used by the *Chant* synthesizer [RPB84], developed at IRCAM in the 80's. This approach models the impulse response of the filter associated to each formant independently as an exponentially damped sinusoid, called a FOF, whose parameters set the formant's frequency and bandwidth. Then the FOFs corresponding to each formants are generated periodically according to the fundamental frequency (as being excited by an impulse train), and summed together. This approach is thus somewhat similar to the parallel filters architecture proposed by Klatt in [Kla80]. While this approach can generate good quality synthesis, it is however only limited to voiced sounds.

Such approaches are also sometimes called rule-based synthesis, as pre-defined rules are required to set the different properties of the voice timbre, or the frequencies and bandwidth of the formants for each vowels, and the way they interpolate from one phoneme to another, for instance. However, it is a fastidious work to find and implement rules in order to generate good quality synthesis over a wide range of vocal timbres and expressions, with all the complex articulations of human voice for pronouncing words. This approach is thus not very flexible for the general case of Text-To-Chant synthesis, but is nevertheless useful for conducting research in an analysis-by-synthesis paradigm, for verifying certain hypothesis that can be difficult to validate using only direct analysis on recordings.
Formant synthesis is also especially suitable for real-time synthesis, as very low latency can be achieved.

2.3.2 Physical modeling synthesis

Physical modeling synthesis, also sometimes called articulatory synthesis, is based on numerical models of voice production and on solving the underlying physical equations, as opposed to spectral models like formants synthesis which are more focused on the principles of sound perception, directly modeling the resulting sound spectrum of the voice rather than the production process itself. Typically, acoustic tube models are used for modeling the vocal tract, and the source is either based on a parametric signal model, a wavetable, or a mechanical model (e.g. a 2-mass model). In physical modeling, a parameter change is directly related to a modification in the voice production mechanism, whereas in spectral models, a parameter change is rather related to a change in the perception.
In such systems, the control parameters are for instance the vocal folds tension, the sub-glottal pressure, or the position of the various articulators (e.g. the jaw, the lips, the tongue, ...) that shape the vocal tract. Examples of such system are the SPASM [Coo89; Coo93], Vox [Kob02], or VocalTractLab [Bir07] synthesizers.
This type of synthesizer is especially suitable for pedagogical purpose, in order to better understand how the voice production works. For instance, the SPASM system has a graphical interface allowing to modify the parameter values and hear the resulting sound in real-time. However, the mapping between the input parameters and the produced voice is quite complex and not very intuitive for controlling the

[3] https://cantordigitalis.limsi.fr/

synthesis, as not directly related to the perception, and no fully-developed TTC system based on this approach exists yet.

2.3.3 Concatenative synthesis

Units selection-based concatenative synthesis has been a very successful approach for speech synthesis, yielding until today the best quality results. It has thus naturally been subsequently applied to singing voice, and the potential high-quality output of this technique motivated us in choosing this approach for the synthesis system developed throughout this thesis, as will be described in chapter 3.
It consists in selecting voice samples in a pre-recorded and pre-annotated database, according to a given input text, and concatenating them to recreate new words and sentences. In terms of intelligibility and naturalness, the main advantage of this technique is that it allows to preserve a timbre as close as possible to that of the original speech signal, especially for timbre variations that naturally occur between phonemes due to the co-articulation effects, which are otherwise difficult to reproduce with previously exposed techniques such as formants or articulatory synthesis.
For best results in speech synthesis, non-uniform units lengths can be used, selected in large databases covering a wide variety of contexts, so that the least processing is applied to those units (ideally no transformation at all) [CB97]. In such systems, units durations can typically go from short diphones to full words, or even sometimes chunks of sentences covering several words. The system should then find the best compromise between selecting the units that best match the target context according to the input text, and selecting the units that can best be concatenated without yielding audible artifacts. This is achieved by computing a target and a concatenation cost for each unit, such as explained in [ZTB09], and then using algorithms such as the viberti algorithm [For73] to find the sequence of units yielding the lowest overall cost among all possible sequences [Vep04].

The target cost computes the difference between the target contextual informations of the sentence to be synthesized and the contexts of the original unit in the recorded database. The definition of a target cost between a candidate unit, u_i, and a required unit, t_i, is

$$C^{(t)}(t_i, u_i) = \sum_{j=1}^{p} \omega_j^{(t)} C_j^{(t)}(t_i, u_i), \tag{2.19}$$

where j indexes among all used contextual informations, $C_j^{(t)}$ is a value (or distance) defined for context j that reflects how different unit u_i is from t_i regarding the context j, and the ω_j are weights set on each context to emphasize the ones that seem more important. Both prosodic and phonetic contexts can be used. Examples of contexts are: the identity of the current, preceding, and succeeding phonemes, the number of phonemes into the syllable, the position and number of syllables into the sentence, etc...
The concatenation (or join) cost between 2 consecutive units u_{i-1} and u_i is defined as:

$$C^{(c)}(u_{i-1}, u_i) = \sum_{k=1}^{q} \omega_k^{(c)} C_k^{(c)}(u_{i-1}, u_i), \tag{2.20}$$

where k can index various spectral and acoustic features, as exposed in [Vep04]. Typical features used are the MFCCs or the f_0.

The sequence of units that minimizes the overall cost (which can be retrieved using the Viterbi algorithm) is then:

$$\hat{u}_{1:n} = \underset{u_{1:n}}{\operatorname{argmin}}\{C(t_{1:n}, u_{1:n})\}, \tag{2.21}$$

where

$$C(t_{1:n}, u_{1:n}) = \sum_{i=1}^{n} C^{(t)}(t_i, u_i) + \sum_{i=2}^{n} C^{(c)}(u_{i-1}, u_i). \tag{2.22}$$

Many concatenation-based Text-To-Speech (TTS) systems exist, among which the Festival Speech Synthesis System [TBC98; BTC01] or the MBROLA system [Dut+96]. In some systems, the concatenation cost may also be replaced by a context clustering approach using decision trees [ZTB09].

With this approach, the larger the database, the more contexts can be covered, and thus the lower the costs are. But for singing, larger ranges of pitch, speed, and intensity are required, as mentioned in section 1.2.1, that can't possibly all be covered in the database. Signal transformations are thus necessary in order to apply the target melody, durations, intensity, and expression. Even thought they should be minimized thanks to the units selection process, spectral discontinuities also exist at junctions between segments, that have to be smoothed by means of signal processing.
Several examples of concatenative TTC systems exists. The main differences between those systems are related to the constitution of the database and unit-selection strategy on one hand, and the underlying signal models used for transforming the units on the other hand.
A first example is the Burcas project [Une02], that is based on diphones concatenation with the MBROLA synthesizer. A second example is the Lyricos system, that uses variable-size units of 1 to 3 phonemes selected by means of decision trees from a phonetically-annotated database (500 made-up words recorded on 2 pitches), and a sinusoidal model with the ABS/OLA method for the signal manipulations, as described in [Mac+97a]. A 3^{rd} system is the Vocaloid software from Yamaha [KO07], which is probably the most popular commercial singing voice synthesizer. This software is based on researches from the Music Technology Group at University Pompeu Fabra, presented in [BL03; Bon08a]. The systems presented in those articles make use of the SMS (Spectral Modelling Synthesis) approach and the EpR (Excitation plus Resonance) voice model - which is based on an extension of the source-filter model - for the signal modeling (see section 2.4), and a database containing steady-states and phonetic articulations (covering at least all diphones necessary for the language to be synthesized, possibly on several pitches).

2.3.4 HMM-based synthesis

Similarly to units concatenation, HMM-based synthesis has first been developed for speech [ZTB09; Yos+99], before being adapted to singing [Sai+06; Our+12; Nak+14]. Similarly to formants synthesis, HMM-based synthesis is based on a source-filter approach, except that the parameters of the source and filter are

obtained from statistics instead of pre-defined rules.
Spectral features and excitation parameters are extracted during a training step from a database of singing recordings, along with some contextual informations, and some statistics are built for each feature from the extracted values. Then, at synthesis, those statistics are used for choosing the best values of each feature according to the target context, clustered using decision trees. With this approach, the filter and source parameters are modeled simultaneously by means of context-dependant Hidden Markov Models (HMMs) [ZTB09]. The source is typically modeled as a simple impulse train, requiring only the f_0 value as a parameter, and the spectrum is usually reconstructed based on the MFCCs (Mel-Frequency Cepstral Coefficients) [Yos+99] and their 1^{st} and 2^{nd} derivatives.
Two examples of singing synthesizer based on this method are the Sinsy[4] [Our+10] and Cevio[5] softwares.

The quality obtained with this type of systems is nowadays still below that of concatenative synthesis in term of naturalness of the voice. One of the main limits of this approach is the limited quality of the vocoder used, the glottal source being often model as a simple impulse train, which introduces some buzziness into the synthesized sound, although some recent works have been conducted to alleviate this problem by using more accurate glottal source modeling [LDR10; Rai+11]. Another drawback is the over-smoothing of the spectral features occurring due to the statistical representation and contexts clustering, as described in [ZTB09]. The quality of this type of systems is also highly dependant on the database used, that should ideally cover all possible contexts in a balanced way, which is difficult for singing due to the high number of independent parameters to be considered (lyrics, pitch, intensity, rhythm, phonemes durations, vibrato, voice quality, ...) and thus requires big databases.

However, one particular advantage of such statistical parametric approach is its flexibility for changing the global voice characteristics simply by modifying the statistics learnt before the synthesis (e.g. to interpolate between different voices), without need for complex signal processing to transform the sounds as in concatenative approaches.

2.3.5 Neural Network based synthesis

The use of deep learning approaches has been growing fast in the last few years in the speech community for many applications, as in many other fields, and singing voice synthesis is not an exception as several neural network-based approaches have recently been investigated for this purpose.
In [Nis+16], the authors describe the implementation of such a system. The architecture of the proposed framework is quite similar to that of HMM-based systems, except that the statistics of the spectral and excitation parameters used for the synthesis are induced by deep neural networks instead of decision tree-clustered context-dependent HMMs. The authors of this study claimed that this method significantly improves the quality of the HMM-based approach thanks to a better prediction of the MFCCs by the system. But as a corpus-based approach, such a system suffers from the same limitations than the HMM-based approach, related to the data sparseness in the learning database, especially for the pitch coverage.

[4] http://www.sinsy.jp/
[5] http://cevio.jp/

Another example with a different architecture, focusing on modeling the spectral envelope only, is presented in [BB16a].
As a last example, [BB17] presents a system for singing synthesis, based on a modified version of the WaveNet architecture [Van+16], that models the features of a parametric vocoder (separating the influence of pitch and timbre) instead of the raw waveform.

2.3.6 Speech-to-singing systems

Although it can't be really considered as synthesis, another type of related systems are speech-to-singing synthesis systems, where some pre-recorded speech signal reading some lyrics is transformed according to a given score to give it a specific melody, rhythm, and some expression [Sai+07]. For this purpose, a segmentation into phonemes of the pronounced sentence is necessary. This kind of system can be useful in an analysis-by-synthesis approach, in order to investigate characteristic features specific to singing voice, without requiring a fully developed synthesis system and while avoiding possible artifacts du to the synthesis process. For instance, in [Roe+12], this approach has been used in order to study the behaviour of the pulse shape parameter R_d of the LF model during vibrato.

2.4 Signal models and transformations techniques for voice processing

As mentioned in the previous section, synthesis systems can be based on different signal modeling and transformation techniques, that have important implications in the flexibility of the system and the achievable quality, for both concatenative and statistical-based synthesis approaches. Signal modeling frameworks can be divided into general purpose approaches that can be used for processing various kinds of sounds, and voice-specific models, also called "vocoders". Some approaches also work purely in the time domain, while other work in the time-frequency domain. This section presents the main state-of-the-arts models and techniques used in speech and singing synthesis systems, in both of those categories. The main transformation required for singing synthesis are pitch-shifting and time-stretching. But some approaches allow for more advanced transformation possibilities than other. Frequency-domain algorithms give more control over amplitudes and phases of the various components and thus usually allow for more flexibility for advanced signal processing (timbre modifications) in singing voice synthesis. We don't aim here at giving an exhaustive review of all the existing models, but rather to demonstrate the diversity of possible approaches, presenting the mains ones, and especially those that have been used during this thesis, along with some of their advantages and drawbacks.

2.4.1 Time-domain approaches (the "OLA" family)

Time-domain approaches assume that the signal can be manipulated without being modeled. The basic technique is to cut the signal into equally-spaced overlapping windowed frames that can then be individually manipulated before summing back up all windows to obtain a new signal. This procedure of summing overlapping windows together is called Overlap-Add (OLA) and gave birth to a family of approaches based on this principle to apply pitch-shifting and time-stretching transformations to sound signals. For time-stretching, the windows can be moved

from their original positions to new positions according to the given time-stretching ratio. Transposition can be obtained by simply resampling the signal after a first time-stretching step to recover the same duration with a different pitch.
However, this simple approach breaks the phase coherence between overlapping windows, which generates some artifacts, and does not allow to preserve the original timbre after transposition. A first improvement is the SOLA algorithm (for Synchronous Overlap-Add) [RW85; ME86]. In SOLA, the new positions of the windows are corrected by a time-lag to overlap the windows at the point of maximum similarity (cross-correlation) within an interval.
A further improvement over SOLA is the PSOLA algorithm (for Pitch-Synchronous Overlap-Add) [MC90; VMT92], which is especially suited for voice processing. In PSOLA, 2-periods-long windows are centered around estimated "pitch-marks" which are assumed to be spaced by one single period of the signal (with fixed period for unvoiced parts). Then, target windows positions are set according to the time-stretching and transposition ratios and mapped to the original windows which are then displaced and possibly dropped or repeated when necessary, before being overlap-added at their new positions, to obtain the desired pitch and durations. Based on the source-filter model, a pitch-synchronous window of speech is assumed to contain a glottal cycle convolved with the impulse response of the vocal tract. As the PSOLA approach doesn't require a resampling step, the timbre (formants structure) of the voice, that is related to the vocal tract's impulse response, is thus preserved after the transformation, which gives a better quality than previous techniques that doesn't allow such timbre preservation. Formants shifting without altering the pitch can also be obtained using PSOLA by resampling each window individually. The PSOLA algorithm has been widely used in the speech community, e.g. for diphone speech synthesis [HMC89], or more recently in various singing voice synthesis systems [Lai07; ABS09].
The MBR-PSOLA approach [DL93] offers further improvements over PSOLA in the case of diphones concatenation-based synthesis, that discard the need for computing pitch-marks while allowing a smoothing of spectral envelope mismatches at diphones' junctions using a simple time-domain waveforms interpolation. The MBROLA [Dut+96] system is used in the "Burcas" [Une02] singing voice synthesizer.
Compared to some of the models presented here-after, these time-domain approaches may be more robust in the sense that they don't rely on advanced signal analysis (except for finding the pitch-marks) that may cause artifacts due to estimation errors. On the other hand, the fact that the signal is not modeled limits the possible transformations, as many voice parameters are not accessible (e.g. source parameters for modifying voice quality).
In [ML95], the authors also presented such time-domain approaches along with other frequency-domain approaches based on the phase vocoder, introduced here-after. Note that a frequency-domain variant of PSOLA (FD-PSOLA) also exists [MC90] to apply transposition, by resampling the signal along the frequency axis rather than the temporal one, or possibly repeating or eliminating some frequency regions.

2.4.2 General purpose models

2.4.2.1 The phase vocoder and superVP

The phase vocoder is an algorithm for processing signals in the time-frequency domain. The original version of the phase vocoder [FG66; Por76; Cro80] is based on a simple Short Term Fourier Transform (STFT) and a phase unwrapping process, without making any assumption about the type of sound being processed or the nature of each frequency bin. The phase vocoder is thus in that sense rather an algorithm offering a general framework for processing sounds than really a signal model. For applying some sound transformations such as time-stretching and pitch-shifting, the frames from the STFT can be moved in time similarly to what is done in PSOLA, or the bins can be shifted in frequency, and the phases of each bin are then corrected according to the computed original unwrapped phases and the transformation's parameters. The phase unwrapping process consists in computing the real phase difference of each bin between successive frames, possibly outside of the $[-\pi;\pi]$ range, based on the bin's frequency. For signal's transformations, the phases are adjusted and wrapped back to the $[-\pi;\pi]$ range. The signal is then re-synthesized by an inverse Fourier transform of the modified frames and overlap-added in the time domain. Some explanations of those transformations possibilities with implementation examples can be found in [Zöl11] (chapter 7).
In the basic version of the phase vocoder, each frequency bin is processed independently. But when a sound contains sinusoidal components, they are usually spread over several bins. This independent processing of the frequency bins can thus lead to some phase de-synchronisation of the bins belonging to a same sinusoid, which leads to artifacts known as "phasiness". To avoid this problem, several improvements have been proposed, with the phase-locked vocoder [Puc95], or other approaches as proposed in [LD97], to keep the phase synchronisation between adjacent bins belonging to a same sinusoidal component (e.g. by applying the same phase shifts to adjacent bins).
Those improvements are efficient to reduce the phasiness due to intra-sinusoid phase de-synchronisation in most sounds. But small frequency estimation errors can also lead to inter-sinusoids phase de-synchronisations. Although this might not always be very audible, the vertical phase alignment between harmonics is especially important for voice, as it relates to the impulsiveness of the glottal pulse shape. An inter-harmonic phase de-synchronisation might thus degrade the perceived quality of the voice. A solution to this problem has been proposed with the Shape-Invarient Phase vocoder (SHIP) algorithm in [Röb10]. SHIP is somewhat similar to a SOLA [RW85] algorithm in the frequency domain. Instead of adapting the phases independently for each partial, which causes their de-synchronisation, the time of maximum cross-correlation t_{x_corr} between 2 succeeding frames is found, and a phase offset corresponding to the time lag between the current frame time and t_{x_corr} is then added to each sinusoidal component, without moving the frame itself as would be done in the SOLA algorithm. The phases being all modified using a similar delay, as a block, the original wave-shape is preserved. One particular advantage of SHIP over SOLA is that the cross-correlation is computed only on the sinusoidal components, without taking the noise part into account, which might otherwise degrade the result. This approach also allows to preserve well the pulse-synchronous amplitude modulation of the stochastic component related to the glottal opening [Röb10]. An evaluation comparing the SHIP and PSOLA algorithms for time scale and transposition

transformation is presented in [Röb10], that resulted in better perceived sound quality for the SHIP algorithm in most cases, with the advantage of using a more flexible frequency-domain approach for more advanced processing. The SHIP algorithm is implemented into superVP.

SuperVP[6] is a software developed at IRCAM in the analysis/synthesis team [LR13] (compiled as a command-line tool), which is based on an extended phase vocoder implementation, integrating a sinusoidal model and many algorithms allowing for various signal analysis and high-quality transformations. Examples of possible transformations are time-stretching with transients preservation (to avoid transient smearing artifacts due to time-stretching with the original phase vocoder), pitch-shifting with envelope preservation [RR05a], filtering and cross-synthesis, and many more. Examples of accessible analysis are f_0 and spectral envelope estimations using various algorithms. Although the phase vocoder is a general purpose algorithm, SHIP is mainly meant for voice processing and superVP has been used in this thesis as the core back-end for the SVP synthesis engine that has been implemented as part of our singing synthesizer ISiS, as will be described later in section 3.5.1.

Similarly to OLA-based approaches, the phase vocoder approach allows time-stretching transformations, with a good quality due to the phase synchronisation process. As in time-domain approaches, transposition can be obtained by resampling the signal after applying a first time-stretching step. Another possible approach for transposition using the phase-vocoder, presented in [LD99a], is to identify the spectral peaks in the STFT and shift the region around each peak to new frequencies, depending on the transposition factor. The frequency-domain implementation in superVP first computes a sinusoidal model and then applies the transposition by modifying the parameters of this model before resynthesizing the sinusoids directly in the spectral domain. The advantages of this frequency-domain technique is that the computational cost is independent of the transposition factor, and that it allows more exotic effects such as non-uniform frequency-dependent pitch shifting or harmonizing.
Transposition with timbre preservation can also be obtained by means of pre-warping the envelope of the STFT frames before the pitch shifting takes place [RR05a]. The pre-warping can be done using a simple amplitude multiplication with factors computed to compensate the transposition of the envelope that will result from the pitch shift. The spectral envelope can be obtained using one of the algorithms presented in section 2.2.3 (e.g. the True-Envelope).

However, this approach still has several drawbacks. A first problem is that when transposing the pitch upward, the low-frequency harmonics are pushed up toward higher frequencies, which may sometimes result into a buzzy sound. A way to circumvent this is to randomize the phase above the original Voice-Unvoiced Frequency (or VUF) to reduce this effect [Röb10]. The VUF is the frequency above which the noise level is greater than that of the sinusoids. Conversely, when transposing pitch downward, the high-frequency content is moved towards lower frequencies and the signal becomes more band-limited. The noise component that is especially prominent in the high frequencies may also be transposed into formants and regions where it is usually not present and thus become more audible,

[6] http://anasynth.ircam.fr/home/english/software/supervp

which increases hoarseness. A solution would thus be to limit the transposition of the noise component, and to create new sinusoids in the high-frequency region for downard transpositions. But while noise can be created from sinusoidal content by randomizing the phase, the inverse process is more tricky. Upward transpositions thus usually result in better quality than downward transpositions. Another important reason for this, when using timbre preservation, is that the higher the pitch, the more sparse the harmonic sampling of the spectral envelope. As explained in section 2.2.3, it is thus more difficult to have a good estimation of the envelope for high pitches, and the timbre may thus be distorted when the harmonic sampling becomes more dense, for downward transpositions. But this problem is inherent to any transposition algorithm using envelope preservation, and not specific to the phase vocoder approach.

2.4.2.2 Sinusoidal models

Sinusoidal models represent sounds as a summation of sinusoids with time-varying frequencies and amplitudes [MQ86]. But, as has been seen previously in section 2.2, voice signals are made of a deterministic (sinusoids) and a stochastic (noise) component, that has to be represented as well. For such sounds, sinusoidal plus residual models have been built to include the noise component. After a first analysis stage, the parameters of the model can be modified to change the pitch and duration of the modeled sound. However, those methods are not specific to voice, as they could be used with other sounds that fit these models (e.g. some musical instruments like clarinet or violin). We shortly review here a few examples of such models.

- **SMS modeling:**
  SMS (for "Spectral Modeling Synthesis") [SJ90] is one such technique that models sounds as a collection of sinusoids controlled by piecewise linear amplitude and frequency envelopes and a time-varying filtered noise component. An analysis procedure first extracts the sinusoidal trajectories by tracking peaks in the signal's STFT. These peaks are then removed and the remaining "noise floor" is modeled as white noise through a time-varying filter.
  This technique has been used in several voice processing systems, among which [Can+00] for voice morphing in a karaoke application, or for singing voice synthesis [Bon+01a; Bon08a], where the SMS analysis is used as a basis to the EpR voice model, as will be described in 2.4.3.2.
  However, as explained in [BS07], the SMS has a similar phase synchronisation issue as the one described for the phase vocoder.

- **HNM:**
  For speech signals, the HNM approach (for "Harmonic plus Noise Model") [Sty01] separates the signal in 2 frequency bands, using an harmonic part to represent the deterministic source component in the lower band, plus a modulated noise component for the non-periodic part in the upper frequency band above the VUF (while the lower band is assumed to contain only harmonics). The noise is modeled using an amplitude -modulated Gaussian noise filtered by an all-pole envelope.

- **QHM, aHM, and aQHM:**
  In order to take into account the fact that the spectral components of the voice

are never perfectly stationary, a Quasi-Harmonic Model (QHM) has been proposed for better modeling small irregularities [PRS08]. Similar methods are the aQHM [PRS11] and aHM [DS12] approaches, for "adaptive (Quasi-)Harmonic Model".

- **Wide-band harmonic sinusoidal modeling:**
  Wide-band analysis consists in using a short window containing only 1 or 2 periods of the signal to perform the analysis, as is done for instance in PSOLA, contrarily to the other previously mentioned methods that tend to use longer windows. While the frequency resolution is lower, such analysis gives a better temporal resolution. The aim of the approach presented in [Bon08b] is to combine a good temporal resolution with the flexibility of frequency-domain methods. This is achieved by means of using a rectangular window of exactly 1 period in the analysis (based on an f_0 analysis), in which case each bin of the FFT corresponds to one harmonic, to estimate the sinusoidal parameters. Time-scale transformations can be applied similarly to PSOLA by repeating, removing or interpolating frames, and transposition or timbre transformations can be obtained by manipulating the harmonic parameters in the spectral domain. This approach has been used in [Bon08a] (where it is referred to as WBVPM, for "Wide-Band Voice Pulse Modeling"), and [BB16b] in the context of singing voice synthesis.

2.4.3 Voice-specific models

2.4.3.1 STRAIGHT

STRAIGHT (for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") is a popular framework for speech processing (analysis, transformation, and re-synthesis) [Kaw97; KMD99], which is widely used in the speech synthesis community (being freely available). In particular, STRAIGHT has been used in [SUA02; SUA05; Sai+07] for synthesizing singing voice.
STRAIGHT re-synthesizes a voice from its analysis using a source-filter approach, from the estimated f_0 values and spectral envelope. The central idea of the proposed method is that it considers the periodic excitation of voiced speech to be a sampling operation of a surface $S(\omega, t)$ in a 3-dimensional space defined by the axes of time (t), frequency (ω), and amplitude. The problem of spectral envelope estimation is thus seen as a surface reconstruction problem using the partial information obtained from the sampled surface. The main goal of the STRAIGHT approach is to remove any trace of interference due to the periodicity, in time and frequency domains, of the voiced speech signal. For this purpose, it is proposed to use a particular f_0-adaptive windowing process to reduce temporal modulations, and further interpolate the harmonic peaks of the speech spectrum along the frequency axis, using 2^{nd} order B-splines as smoothing functions. This 2-step procedure leads to a smooth spectrogram representation.
The synthesis engine then consists of an excitation source and a time-varying filter. This time-varying filter is implemented as the minimum phase impulse response calculated from the smoothed time-frequency representation, while the source is modeled as shaped pulse and noise based on phase manipulations.

2.4.3.2 The EpR model

While the STRAIGHT system has been thought for speech processing and has thus been classified here as a voice-specific model, it is not based on a physiological model of voice production, and could thus be still applied to other kind of harmonic sounds. Especially, it assumes a flat source spectrum and thus does not allow to treat the glottal source and vocal tract filter as independent components.
The EpR voice model [Bon+01a; Bon+01b; Bon08a] (for "Excitation plus Resonance") is based on a more physiologically motivated extension of the simple source/filter model. It is built, from real voice sounds, on top of the sinusoidal plus residual representation, obtained by the SMS analysis mentioned above [SJ90], to decompose and parametrize the voice spectrum as independent perceptually relevant components, useful for transformation purposes. It models the magnitude spectral envelope as 2 filters in cascade, plus a differential spectral shape envelope. The first filter models the voice source frequency response using an exponentially decaying curve plus one resonance in the low-frequencies to model the glottal formant. The gain and slope of this curve are obtained from a regression on the harmonic peaks. The second filter models the vocal tract as a set of resonances which emulate the voice formants. The source and vocal tract resonances are modeled as second order filters (based on the Klatt formants synthesizer [Kla80]), implemented in the frequency domain.
In the EpR framework, two models are considered for the harmonic and residual components, to allow independent modifications, that share the same resonances. Thus, the input to the filters are an harmonic comb in the frequency domain, and a voiced residual excitation obtained from the residual of the SMS analysis. The excitation for the unvoiced parts of the sounds uses directly the original recording of the singer.
The phase alignment of the harmonics is obtained from the EpR spectral phase envelope, which assumes that each filter resonance (except the source one) produces a linear shift of π on the flat phase envelope.
This model allows to approximate the voice spectrum, but is not a perfect fit (especially it does not model the anti-resonances of the vocal tract). For a more accurate modeling, the differences (in dB) between this model and the real harmonic envelope is thus added to the modeled spectrum.
For resynthesis, the EpR model is converted back to SMS parameters. This model has been implemented in a singing voice synthesizer, as explained in [Bon+01a; Bon+01b; Bon08a].

2.4.3.3 Parametric LF-based voice models: SVLN and PSY

SVLN:
Similarly to the EpR model, SVLN [Deg10] (for "**S**eparation of the **V**ocal-tract with the **L**iljencrants-Fant model plus **N**oise") is a voice-specific analysis/re-synthesis method based on a physiological model of voice production, that explicitly models the glottal source and VTF as independent components.
Based on the source-filter model, as presented in section 2.2, SVLN separates the voice spectrum $S(\omega)$ into 4 components: a deterministic source $G^{R_d}(\omega)$ representing the glottal pulse, a noise source $N^{\sigma_g}(\omega)$ representing the stochastic noise component, the VTF $C^{\overline{c}}(\omega)$, and the lips and nostrils radiation filter $L(\omega)$. In the proposed method, the LF model parametrized with R_d is used to model the deterministic component of the glottal source $G^{R_d}(\omega)$, and zero-mean gaussian noise

with standard deviation σ_g represents the noise component $N^{\sigma_g}(\omega)$. Then, in an analysis step, the VTF is estimated by taking into account this source model to fit an observed speech spectrum. For a fully parametric model, The VTF $C^{\bar{c}}(\omega)$ is finally represented by its cepstral coefficients $\bar{c}$.
Using those components, the sound spectrum is modeled as:

$$S(\omega) = [e^{j\omega\phi} \cdot H^{f_0}(\omega) \cdot G^{R_d}(\omega) + N^{\sigma_g}(\omega)] \cdot C^{\bar{c}}(\omega) \cdot L(\omega) \tag{2.23}$$

where $H^{f_0}(\omega)$ is an harmonic comb with fundamental frequency f_0, and $e^{j\omega\phi}$ is a simple delay related to the temporal pulse position.

Analysis:
The SVLN method models the source component using four parameters (f_0, R_d, E_e, σ_g). The fundamental frequency f_0 may be estimated using various existing methods (e.g. [CK02; CH08; KAZ16]). The LF shape parameter R_d can be estimated using the phase minimization method introduced in section 2.2.4. The mean log amplitude of the VTF is arbitrarily fixed to zero so that the energy variation of the speech signal is only dependent on the energy of the source (i.e. the excitation amplitude of the glottal model E_e and the noise level σ_g). σ_g is computed as the source spectral amplitude at the estimated Voiced/Unvoiced Frequency (VUF) F_{VU}: $\sigma_g = \left|G^{R_d}(F_{VU})\right|$.
Based on these estimations, the VTF can be extracted from the speech signal. In the deterministic band, where $\omega < F_{VU}$, the contributions of $L(\omega)$ (=$j\omega$) and $G^{R_d}(\omega)$ are removed from $S(\omega < F_{VU})$ by spectral division. In the upper noisy band, where $\omega > F_{VU}$, $S(\omega)$ is divided by $L(\omega)$ and by the value $\left|G^{R_d}(F_{VU})\right|$ to ensure a continuity between the two frequency bands. (Note that on unvoiced segments $F_{VU} = 0$.) Then, the True Envelope (TE) algorithm is used to fit the envelope on the division result, and the estimated VTF is finally converted to cepstral coefficients $\bar{c}$.

Synthesis:
The synthesis procedure of SVLN is basically an overlap-add technique. Small segments of stationary signals are synthesized and these segments are then overlap-added to construct the whole signal. For synthesizing the deterministic part, temporal marks m_k are placed at intervals depending on the fundamental period $\frac{1}{f_0}$, and a glottal pulse is generated around each mark, its maximum excitation instant t_e corresponding to m_k. For the stochastic component, a noise segment is also generated and centered on each mark m_k. In unvoiced parts, segments of 5*ms* duration are used. If the generated noise is white, the synthesized voice sounds hoarse because the lowest harmonics of the deterministic source are disturbed by this noise. The noise is thus filtered with a high-pass filter $F_{hp}^{VUF}(\omega)$ defined by a cut-off frequency equal to the VUF and a slope of 6 dB/kHz in the transition band. Furthermore, in voice production, this random noise component $N^{\sigma_g}(\omega)$ originates from air turbulences generated at the glottis. It thus also needs to be modulated in amplitude synchronously with the fundamental period, according to the LF pulse shape, otherwise it is perceived as a second source separately from the deterministic source. This noise component is then cross-faded between consecutive segments.
Then, the deterministic and stochastic source components can be mixed, and the VTF and radiation filters are applied following equation 2.24:

$$S_k(\omega) = (e^{-j\omega m_k} \cdot G^{R_{d_k}}(\omega) + N_k(\omega)) \cdot C^{\bar{c}_k}(\omega) \cdot j\omega \tag{2.24}$$

where $e^{-j\omega m_k}$ is a delay placing the instant te of the LF model at the mark m_k and $C^{\bar{c}_k}(\omega)$ is the minimum-phase VTF corresponding to the cepstral coefficients $\bar{c}_k$. Finally, the time domain sequence of each segment is retrieved through the inverse Fourier transform of $S_k(\omega)$.

The SVLN approach being fully parametric, it is well-suited for being used in HMM-based synthesis systems. Another advantage is that it provides meaningful control parameters directly related to the voice production model, allowing voice quality transformations such as breathiness or tenseness modifications.
However, the Gaussian noise used by this method is not realistic enough and degrades the quality of the generated sound, especially in unvoiced segments.

PSY:
Based on SVLN, the PSY approach (for "**P**arametric speech analysis, transformation and **SY**nthesis"), introduced in [Hub15], has then been developed, aiming at improving the quality given by SVLN in the context of voice conversion. The framework of PSY is basically an extension of the SVLN system, the main improvements being the extraction of the real noise component from the original signal instead of directly using filtered Gaussian noise for modeling the stochastic source component, and the suppression of spectral ripples in the deterministic source spectrum for high R_d values (visible on figure 2.6 b).) that disturbs the proper estimation of the VTF using spectral division. PSY is based on the following voice production interpretation in time domain :

$$s(n) = u(n) + v(n) = u(n) + \sum_i g(n, P_i) * \delta(n - P_i) * c(n, P_i) \tag{2.25}$$

where $u(n)$ and $v(n)$ are respectively the unvoiced and voiced component, $g(n, P_i)$ is the glottal pulse (represented as the glottal flow derivative) related to the GCI position P_i, $\delta(n, P_i)$ is an impulse at position P_i, and $c(n, P_i)$ denotes the VTF at the same position.
In PSY, the unvoiced stochastic component $U(\omega)$ is first extracted from the spectral representation $S(\omega)$ of the original signal s(n). This is done by deleting the sinusoidal content, based on the Deterministic plus Stochastic Modeling (DSM) approach [Ser89]. Similarly to SVLN, the signal processing in PSY is done in the spectral domain, using the STFT. The radiation filter $L(\omega)$ is not explicitly present in PSY, as it is implicitly contained in the glottal flow derivative g and unvoiced component u.
The main problem for separating the sinusoidal and noise components is the fast frequency and amplitude modulations that may occur sometimes, e.g. at voice boundaries. The ReMiDeMo (for "**Re**-**Mi**xing with **De**-**Mo**dulation") method introduced in [Hub15] aims at easing the detection and removal of the sinusoids by first demodulating (flatten) the f_0 and amplitude of the signal. The original f_0 contour is flattened to its mean value by means of time-varying resampling. The varying amplitude contour of s(n) is demodulated by means of dividing the signal by its smoothed Hilbert transform, similarly to [Yan08]. After this demodulation step, the sinusoidal content is subtracted from the signal, and the original modulations are applied back on the residual signal to retrieve the unvoiced residual signal.
The suppression of spectral ripples occurring for high R_d values is done by estimating a smooth spectral envelope on the spectrum of the synthesized deterministic source component.

A similar approach to the SVLN and PSY frameworks has also been integrated in our system *ISiS*, as will be presented in section 3.5.2.

2.4.4 Summary on voice modeling techniques

As we have seen, many different approaches can be used to model and transform voice signals, each having different advantages and limits.
Time-domain approaches like PSOLA are quite simple, requiring only the f_0 (and pitch-marks) estimation, and provide relatively good quality transformations. However, this approach doesn't allow advanced timbre manipulations.
For this purpose, frequency-domain approaches like the phase vocoder and sinusoidal models are more appropriate, as the amplitude and phase of each component can be independently modified, while retaining a maximum of the informations from the original signal to provide high-quality transformations. Especially, the simple STFT-based analysis in the phase vocoder retains all the information and can exactly resynthesize the original signal without any artifacts if no transformation is applied. However, those approaches remain general purpose approaches and don't provide access to voice-specific parameters related to the glottal pulse shape and formants.
Voice-specific models like the EpR, SVLN and PSY frameworks are thus more flexible for advanced voice transformations, allowing to manipulate the source parameters to modify the voice quality (tenseness, breathiness, ...). The EpR model also gives the possibility to control the formants parameters. However, those approaches are also dependant on more complex (and thus less error-prone) analysis steps (R_d, VUF, noise residual, formants, ...), which can create artifacts in the resynthesis.

2.5 Expressive voice transformations

In the previous section, we reviewed various techniques to model voice signals allowing to perform the 2 main transformations that are absolutely necessary for singing voice synthesis, which are pitch and duration modifications, required to match the target melody and rhythm given by the input score. However, in order to improve the naturalness and expressiveness of synthetic voices, it is also necessary to modify the timbre according to the intensity, pitch, and voice quality. For statistical-based and concatenative approaches, all those timbre variations should ideally be included in the system's database in order to be reproduced accurately. However, due to the very wide variety of timbres and parameters to be considered in singing, this task is hardly achievable. It is therefore necessary to design perceptually relevant rules and algorithms to allow such advanced timbre transformations. We review in this section the main types of transformations that would be desirable for generating realistic synthesis in various singing styles, and some state-of-the-art approaches to achieve it.

2.5.1 Intensity

Past studies have permitted to gather knowledge about how various spectral features of voice may change during speech and singing, according to intensity, and various aspects have to be considered. Especially, the intensity of voice is related to the notion of vocal effort, which is, as pointed out in several sources

[PD16; TE00; LB13], physiologically linked to an increase in subglottal pressure, an increase in vocal-fold tension, and a wider mouth opening.
From the view point of the source-filter model, the well-known effect of increasing the vocal folds' tension on the source spectrum is an increase of the glottal formant's frequency and a decrease of the spectral tilt, while the main effect related to a wider mouth opening is an increase of the 1^{st} formant's frequency (F_1), as observed in many studies [PD16; TE00; LB13; LD99b; Hub+99]. The inverse effects are observed when lowering vocal effort.
While these features are easy to modify in formant synthesizers like [Feu+17], where all source and formants' parameters can be explicitly controlled, this task is much more complicated for systems based on signal transformations like [KO07], which require to use specific approaches to transform the sound samples.

A first possible approach to create transformations of vocal intensity is spectral morphing, that uses target templates recorded at different levels of low and high vocal efforts, as proposed in [SG03; Tur+05; DSC13] for diphones concatenation-based speech synthesis. In [SG03], recordings on 3 different vocal efforts (soft, modal, and loud) are used, but the proposed approach doesn't allow to interpolate between them to produce a continuous control of vocal effort from soft to loud, which would be a desirable goal. [Tur+05] and [DSC13] present improvements over this system using morphing with target envelopes to interpolate between the recorded low, modal, and high vocal efforts. In [Tur+05], the interpolation of the spectral envelopes is done using LSFs [KR86], based on an LPC analysis. [DSC13] proposes a similar method, but uses 9^{th} order polynomials instead of LPC for representing the spectral envelope, where this parametric representation is obtained based on an harmonic model (HNM). The advantage of the spectral morphing approach is that it includes both the effects on the source and the vocal tract. However, recordings at different vocal intensities are not always available.

An alternative to spectral morphing is to use a parametric approach to produce such effect without the requirement of additional recordings. In this direction, [ADC98] and [AD03] focused on the source-related spectral characteristics, filtering in the spectral domain to modify the spectral tilt and glottal formant. In [ADC98], the glottal source and VTF are first separated using the IAIF [Alk92] approach, and the spectrum is then modified by multiplying the amplitudes of the harmonics by frequency-dependant factors. The ratio of the voiced and unvoiced components is also modified. [AD03] uses a stylisation of the source amplitude spectrum using 3 linear segments (or asymptotes). Then, the source spectrum can be transformed by modifying the slope of these segments and change the harmonics amplitudes accordingly, as done in the previous method.

But when the spectral tilt is decreased for high vocal effort values, the VUF should increase, which can't be achieved only by amplitudes modifications, as new harmonics should be created in the high frequencies above the original VUF. This can be achieved using certain parametric voice models like the SVLN [Deg10] and PSY [Hub15] approaches previously described, by modifying the R_d parameter. In [Fan97], Fant proposed a rule to coherently modify this R_d parameter coherently with the signal's energy by increasing $\frac{1}{R_d}$ and E_e by steps of respectively 1dB and 2dB.
[PD16] recently proposed another means to add the missing high-frequency harmonics in the spectrum for weak-to-loud transformations of singing voices, using a

wave-shaping approach based on a specific time-domain warping function applied on a rough estimation of the glottal source (similarly to a distortion audio effect). Then, the spectral tilt is modified by adding a certain gain value in dB/decade.

However, modifying only the source spectrum is not sufficient for producing a convincing effect, and the Vocal Tract Filter (VTF) should also change accordingly with intensity. [PD16] thus also modifies the position of the 1^{st} formant by $10H_z$/dB. In [Mol+14], the authors used a parametric model of spectral envelope based on 4-poles resonators to modify the gain, spectral tilt, and formants' frequencies and bandwidths, based on regressions of those parameters computed from 60 vowels recorded at different intensities. The main drawback of this approach is that formants' parameters are not straightforward to estimate and require to be manually corrected. But assuming an appropriate estimation, the transformation can then be easily applied by modifying the parameters like in formants synthesizers.

Some researches about realistic intensity transformations have been conducted in the framework of this thesis, and an algorithm for simulating the effect of mouth opening on timbre has been developed as a first step towards a more complete vocal intensity effect. This work was the subject of a publication [AR17] and will be developed in chapter 6.

2.5.2 Pitch

Modifications of pitch in singing voice is also related to changes in the voice production system, both at the source and vocal tract levels, that have been studied from the physiological and signal point of views.

2.5.2.1 Laryngeal mechanisms

Relating to the glottal source, several vibratory mechanisms can be observed, that are used in different conditions, and associated to a specific pitch range. 4 laryngeal mechanisms can be identified in human voice, each being related to a specific register. The principal mechanisms, that are used most of the time by singers, are the mechanism M1 for chest register, and the mechanism M2 for falsetto register. The M0 mechanism corresponds to the fry voice, and the mechanism M3 to the whistle register (extremely high pitch). These different mechanisms are associated with various parts of the vocal folds (e.g. the whole folds or only the edges) vibrating with different amplitudes, which creates different glottal source shapes with specific characteristics [RH09; Ber+14], and thus can produce different vocal qualities. Note however, that the frequency range of the chest and falsetto registers overlap in an interval where the singer may use either mechanisms. Singers can also learn to extend those ranges to some extent, using specific techniques. Western-European opera singers train particularly to minimise the differences of timbre between the registers, so as to avoid audible timbral discontinuities while changing pitch. At the contrary some other vocal techniques like yoddle extensively use the change in timbre between the chest and falsetto registers to create a characteristic effect [Wis07].

In [Ber+14], the author studied the behaviour of the glottal source and its influence on the vocal tract's resonances for the chest and falsetto registers of male

operatic singers in their overlapping range, using electroglottography (EGG), and found important differences in the glottal contact quotient, as well as differences in the 1st and 2nd formant's frequencies, for these 2 mechanisms. Similarly, [RH09] also studied the characteristics of the different mechanisms using EGG and found differences of shapes between glottal cycles at the transitions between the M1 and M2 mechanisms during glissendi, and especially in the open quotient parameter. Note that, as stated in [Ber+14], the laryngeal mechanism also has an influence on the vocal tract resonances.

In [Feu+17], the author proposed different settings of the glottal source model parameters O_q (open quotient) and α_m (asymmetry coefficient) for the mechanisms M1 (chest register) and M2 (falsetto) in a parametric formant synthesizer. But no dependency on the pitch is given, and the mechanism has to be chosen by the user. Beyond that, not much work has yet been conducted in the modeling or transformation of vocal registers to our knowledge. However, one can assume that the effect of laryngeal mechanism on timbre may be partly implicitly modeled in data-based approaches like HMM-based synthesis and in spectral morphing (similarly to intensity transformations), if the database contains various pitches corresponding to different registers, as the singer in the recordings would naturally use various mechanisms depending on the pitch.

2.5.2.2 Formants tuning

Besides the vibratory mechanism, the resonances of the vocal tract also tend to be adapted together with the pitch. Many studies have investigated this question and showed evidences of specific strategies used by singers to tune the lowest formants' frequencies with the f_0 or higher harmonics (which is often referred as "formant tuning") [HSW11; SLG13; BS00; Gar+10; JSW04].
As explained previously, the problem of estimating the real spectral envelope (and thus formants' parameters) can not be solved easily, especially for high-pitched voices. Several solutions have thus been proposed in order to get reliable estimations of the vocal tract's resonances to investigate this question, among which injecting broad-band noise into the singer's mouth, that is then naturally filtered by the vocal tract during singing, and recording the resulting sound [HSW11; Gar+10; JSW04]. This approach has the advantage to be non-invasive compared to other techniques.

An advantage of tuning the resonances to the harmonics' frequencies is to gain in loudness by amplifying those harmonics and thus the overall sound level [CS92; MS90], so that the voice gets more audible to the audience (e.g. for opera singers singing along with an orchestra).
This formant tuning effect is especially observed on high-pitched soprano voices [HSW11; Gar+10; JSW04], as the f_0 value may more easily go beyond the 1st formant's frequency F_1. The usual observation is that when the f_0 is low, F_1 remains fixed, but when the f_0 get above F_1, F_1 approximately follows the f_0 value (F_1:f_0 tuning). In the lower range, or for other types of singers (altos, tenors, baritones), other tuning strategies may also be used, such as for instance F_1:$2f_0$, or F_1:$3f_0$.
Note that, the vowel's identity being mostly determined by the positions of the 2 1st formants, formant tuning is also a phoneme-dependent phenomenon [MS90;

CS92].
The previously cited studies mainly focused on classically-trained singers. But the strategies used may also depend on the singing style. For instance, [BS00] explored the strategies used for the belting style and showed that in this case the loud and bright sound typical of the belting style is achieved by the implementation of resonance strategies that enhance higher harmonics.
Formant tuning is also extensively used in throat singing, but in this case independently of the pitch, in order to enhance specific overtones and thus create the impression of singing at 2 different pitches at the same time [Smi67; Kob04].

Few means of applying such effects for singing synthesis have been implemented yet. However, similarly to intensity modifications, such rules are rather straightforward to implement in formants synthesizers, as done in [Feu+17]. The perceptual effects of formants tuning had also been investigated in [CS92] using the MUSSE DIG formants synthesizer.
In [San+16], a parametric model of spectral envelope, based on previous work on intensity transformation [Mol+14], has been recently used to achieve natural pitch modifications. Many sung vowels have been analyzed to deduce pitch-dependency rules for the 3 first formants, using linear regressions. Based on the proposed parametric envelope model, pitch-shifting is then applied using PSOLA and inverse filtering to replace the original envelope with the new one, after applying the rules.
In [Don+11], the authors proposed to use Dynamic Frequency Warping to learn the mapping between the spectrums of vowels sung at different pitches, and thus obtain a better quality of pitch-shifting by taking into account the pitch-dependent differences in the spectral envelope, without explicitly modeling rules for formant tuning.

2.5.3 Singer's formant

Another specific attribute present in some singing voice is the "singer's formant". As explained in [Sun90]: *"In western male operatic voices, the third, fourth, and fifth formants tend to cluster together, producing a large peak around $3kH_z$ in the spectral envelope called the "singer's formant", that raises the sound level, thus making it easier to hear the singing voice over a loud orchestra"*. Its level relative to the 1st formant's peak amplitude can vary depending on factors like the vowel, pitch, or vocal loudness, and it is particularly present in professional classically trained singers (bass, baritone, tenors and alto), compared to untrained singers [Sun01].
In [Sai+07] and [Lee+14], the authors propose to simulate the singer's formant, using an band-pass filter centered on the nearest peak of the spectral envelope around 3kHz, for speech-to-singing transformations.

2.5.4 Vocal roughness

In some musical styles, such as pop, blues, rock, jazz, punk, metal, etc..., some specific timbre effects related to vocal roughness can be used expressively [Sak+04; SDB12; Cha13]. However, the term "roughness" can designate a rather wide variety of different voice qualities, also described with terms such as harsh, hoarse, saturated, growl, ... More generally, a rough voice can be defined as a voice that presents some irregularities, or "asperities", that are not present in other more "neutral" voice qualities like modal voice. Vocal roughness is often associated to

a certain level of vocal effort, but a loud or shouted voice does not necessarily exhibit a rough quality, and this aspect should thus be treated separately.
In the spectral domain, a rough voice is characterized by a low Harmonic-to-Noise Ratio (HNR) [Tsa+10; SO84], with the presence of noise and subharmonics (sinusoids present between the harmonics of the voice) [Sak+04; NH02; Nie08; BB13]. In some more extreme examples of vocal roughness, the sound becomes completely noisy and it is not even possible to identify a pitch in the signal [Nie08].
In the time domain, rough voices are mainly characterized by the presence of jitter and shimmer [BB13; VK05; Jon+01]. Jitter can be defined as a period-to-period irregularity of pitch (each pitch cycle may have a different duration), while shimmer is defined as cycle-to-cycle amplitude variations (each glottal pulse may have a different amplitude). In some cases, one can observe some macro-pulses, where a macro-pulse is a group of pulses (with varying shapes and amplitudes) that exhibit a certain periodicity at a lower rate than the real f_0 [Nie08]. Rough voices may also be more or less stable, and in some cases (e.g. in screamed voices) present bifurcations, defined as sudden and uncontrolled transitions between different vibratory behaviours (e.g. different number of sub-harmonics) and possibly to a chaotic regime [Lag+16; NH02; Bai09].

Vocal roughness results from non-linear phenomena in the vocal production system and, depending on the effect, may have various physiological causes that can hardly be determined from the signal itself. Although we are interested in our case in the expressive use of roughness in singing, similar perceptual effects may be found in screams or shouted voices, as well as some pathological voices which may imply similar mechanisms. A first possible cause of roughness are laryngeal mucous lesions (nodules, polyps) or laryngeal mobility lesions (paralysis) [Muñ+03]. Whereas vocal folds are usually coupled, asymmetries between the 2 vocal folds (e.g. in tension) may cause them to vibrate at different frequencies [NH02; Tig+97; Gio+99], which can create roughness. Although these causes are not intentional, some permanent voice disorders may be involved in the identity and expressive quality of certain singers' voices [Cha13]. Besides the vocal folds, other supra-glottal structures, like the ventricular folds, the arytenoid cartilages, aryepiglottic folds, or the epiglottis, may also vibrate and thus be implied in the creation of a rough voice quality [NH02; Tsa+10; Sak+04; Bai09]. Ventricular folds (positioned just above the vocal folds in the larynx, as shown in figure 2.1) are also involved in certain throat singing techniques to generate very low-pitched voices by inducing a period-doubling phenomena [Hen+06; Bai09].
Note that besides the rough voice quality, the physiological mechanisms involved (e.g. supra-glottal constrictions, or rising of the larynx) may also change the resonances of the vocal tract that have implications on the overall voice timbre [Bai09], which also present a certain degree of roughness.

From the perceptual point of view, the perception of roughness is related to the ability of the auditory system to perceive and resolve individual sinusoids presented together, as explained in [Sun90]: *"The condition for our ability to hear one of 2 equally strong spectrum partials as an individual tone is that they are separated by at least one critical band. [. . .] All pairs of partials that are similar in amplitude and separated by less than a critical band contribute to the roughness of the timbre. If the pair of partials is high in amplitude, the contribution is substantial."* This explains for instance why the presence of sub-harmonics in the voice spectrum is perceived as roughness. Similar conclusions were drawn

in [Ter74] from the study of amplitude and frequency-modulated tones, which adds that *"the entire roughness is composed of the partial roughnesses which are contributed by adjacent critical bands"*.

As vocal roughness covers multiple different voice qualities, different approaches may be used to model different voice qualities. In [Nie08], the author classified roughness-related vocal effects found in different singing styles, denoted as "Extreme Vocal Effects", into 5 categories (rattle, distortion, growl, grunt, and scream, from the softest to the more extreme), giving a description and musical references for each. Based on the Wide-band Harmonic Sinusoidal Modeling algorithm [Bon08b], the author tried to reproduce those effects, from the analysis of recordings, by a combination of various treatments: global stretching of the spectral envelope; spectral filtering to introduce macropulses (using a different filter for each glottal pulse); addition of noise based on phase randomisation; addition of pitch variations (jitter); and a negative gain on the fundamental. Each one on those parameters has a different setting, depending on the effect to be produced. The "growl" effect, characteristic of blues music such as employed by Louis Armstrong, is probably the most popular of these effects.
In [LB04], the author proposed 2 approaches to create such effects. The 1st approach consists in transposing down the original signal by a certain number of octaves, then shift and scale several copies of this transposed signal with various delays and gains values with a certain amount of randomness (to introduce jitter and shimmer), and finally sum them together.
The 2nd approach presented in [LB04] consists in adding sub-harmonics to the signal directly in the frequency domain, based on the phase vocoder. In this approach, sub-harmonics are added only in the range [f_0-8kHz]. The phase and amplitude patterns of these sub-harmonics are imposed based on the analysis of real growl sounds.
Another frequency-domain approach is presented in [BB13], where authors proposed to use spectral morphing to mix an original "clean" voice to be transformed with a sample of rough voice with the desired voice quality. This is achieved by first inverse filtering the rough sound by its spectral envelope to get a residual signal, apply the target f_0 curve by time-domain resampling, filtering it back with the spectral envelope of the original clean sound, and finally transforming the harmonics in order to match the phases and amplitudes of the original sound. The rough source can also be looped if necessary to match the target duration.
Other approaches are based exclusively on jitter and shimmer modeling, for transforming speaking or singing voices. In [VK05], jitter is defined as the average intensity in a band around the fundamental in the spectrum of a normalized pitch contour. The jitter can thus be introduced or modified by changing the mean and variance of the energy in this band.
In [RL08], a generative model based on statistical analysis of natural hoarse voices is used to modify the jitter and shimmer properties of a modal (or "clean") voice. The jitter is first obtained by high-pass filtering the f_0 contour. Then some statistics are extracted on the degree of jitter and the numbers of consecutive pitch cycles without alternations of the jitter derivative. "Jitter banks" are built to store the original pitch variations due to the jitter. Then, based on these statistics and jitter banks, a new pitch curve including jitter can be generated and applied by time-domain resampling of each individual pitch cycle obtained by a pitch-synchronous analysis (using envelope preservation).

Two approaches for introducing roughness in singing voice have been investigated during this thesis, which will be presented in chapter 6.

2.5.5 Breathiness

Another attribute of the voice timbre that may be characteristic of some singers or be used punctually as an expressive means is breathiness [Cha13], which manifests itself as high-frequency noise in the speech spectrum. It is due to significant air leakage between the vocal folds when the voice is relaxed, that produces aspiration noise, and is thus strongly related to vocal effort [Nor+08; FRR09]. One particular aspect of breathy voices is also an increased spectral tilt, related to a low vocal effort, which can be obtained using similar techniques to those previously presented for intensity transformations. Modifying the breathiness thus requires to be able to manipulate both the noise level and the tenseness (spectral tilt) of the voice.

In [TD04], the HNM model is used to perform such breathiness transformations of singing voices. As explained in this article, simply boosting the gain of the noise component to increase breathiness is not sufficient and may increase artifacts due to errors in the analysis. Instead of the real noise component, high-passed white noise filtered by the VTF is thus used, as in [Deg10; Hub15].
In [DRR11b], the SVLN analysis/re-synthesis method is used to perform breathiness transformations by modifying the R_d source parameter between the analysis and synthesis stages to increase the spectral tilt. In [Nor+08], the authors also considered the need of scaling the vocal effort in coherence with the breathiness modification in order for the stochastic and deterministic components to blend well together, without perceiving the noise as a separate source. For this purpose, an adaptive "pre-emphasis" filter representing the source's spectral tilt is estimated using linear prediction with a low order. Then, transforming high-effort singing voices into breathy voices is achieved by manipulating the spectral emphasis filter (based on the analysis of breathy vowels) and adding pulsed white noise to simulate aspiration noise.
Similarly, the authors in [Nor+08] performed breathiness transformations by low-pass filtering the original voice to increase the spectral tilt, and adding filtered white noise.

2.6 Expression control

We reviewed so far the basics of voice production and the main state-of-the-art techniques used for modeling, synthesizing, and transforming voices. We will detail in chapter 3 how we implemented such techniques in the framework of a singing voice synthesizer that we developed. But beyond generating an intelligible and natural-sounding voice timbre, similar to that of a real human voice, synthesizers must also reproduce the various expressive intentions and unintentional fluctuations of real singers.

The specificity of singing is that it is at the intersection of speech and music. While speech synthesis and music performance have been studied separately, singing synthesis share common aspects with both fields. Speech prosody has been thoroughly studied from the signal, but also psychological or cognitive point

of views, with respect to naturalness and expressivity, considering aspects such as speaking style and emotions [Bel09; Obi11; Fón83]. Speech prosody can be defined as the behaviour over time of the acoustic features that don't affect the identity of the speaker and the phonemes pronounced, encompassing pitch, intensity, speaking rate and some spectral features (also related to terms like intonation, tone, stress, accent, rhythm, ...), and many different techniques have been proposed to generate appropriate prosodic features for speech synthesis such as reviewed for instance in [Obi11]. On the other hand, some works have been focused on musical expressivity, with the aim to render realistic interpretations of musical pieces, based on general features such as intensity and tempo variations to emphasize some parts of a piece and add some grouping structures (or "phrasing") to the notes of the score, not necessarily focusing on a particular instrument. As said in [WG04]: *"The purpose of computational models of expressive music performance is to specify precisely the physical parameters defining a performance (e.g., onset timing, inter-onset intervals, loudness levels, note durations, etc.), and to quantify (quasi-)systematic relationships between certain properties of the musical score, the performance context, and an actual performance of a given piece."* In the context of the present work, both fields of research are thus useful sources of knowledge and inspiration.

While signal modeling techniques are for the main part very similar for both, speech and singing synthesis differ in their timbral and prosodic characteristics, and generating an expressive singing voice requires an appropriate control of both aspects. While the main purpose of speech is to convey a message (either explicitly in the pronounced text, or implicitly using prosodic features to shape the sound), an important aspect of singing, compared to speech, is that the pitch and timing are constrained by the score being interpreted. Moreover, aesthetic considerations are much more important in singing, which makes use of specific expressive features such as vibrato, portamento, crescendo, etc..., that are not found in speech. For these reasons, we prefer, in the case of singing voice, to talk about "expression control" rather than prosody
Compared to other musical instruments, voice is very flexible. For instance, a piano has a basically flat pitch for each note, with a relatively restricted and stable timbre. The main parameters considered for rendering a piano performance would thus simply be the tempo and dynamics as in [GPW04; WG04]. Those approaches work at the note's and phrase's levels, but do not predict intra-note behaviours which are very important for singing, whose characteristics are continuously evolving in time, especially in terms of pitch and timbre, and which thus require much more parameters to be controlled. A particular aspect of voice compared to other music instruments is also the phonetic content related to the pronounced text, that needs to be adequately controlled in term of timing, which adds a further temporal dimension to be considered.

As already said, the control parameters for singing voice are constrained by the score. But all the subtle variations of the voice can't be explicitly defined in a simple score, which provides only basic symbolic informations, such as the pitch and duration of the notes, and possibly additional informations related to the dynamic (nuances, crescendi, ...) and articulation (e.g. legato or staccato). The score itself is thus not sufficient to render a realistic performance, and the addition of natural fluctuations and specific expressive variations is necessary. The success

of conveying a natural expression to the synthesis relies on an exhaustive and coherent control of all the various acoustic features, taking into account their possible inter-dependencies. The synthesizer thus requires a means to generate adequately those features, automatically from the input score and text. This is the purpose of the control module (illustrated in figure 1.1), which makes use of explicit and/or implicit knowledge to achieve this task, by means of specific models, rules, and/or machine learning techniques, possibly based on a set of reference performances recordings. Singing synthesizers having nowadays achieved a reasonable quality in terms of signal modeling (especially using concatenative approaches), more research has been recently dedicated to the problem of expression control. But progress are still necessary to obtain a singing quality comparable to that of a professional singer, for a wide range of singing styles.

In this section, we will first review the various control parameters related to singing voice and their principal characteristics. Then we will present the state-of-the-art techniques used for generating such parameters.

2.6.1 Control parameters

Many control parameters can be identified to cover all aspects of singing synthesis. These parameters confer both a natural and expressive character to the voice, covering features related to emotion, singing style and inter-individual variations, while carrying the melody and rhythm imposed by the score.
These parameters are of different natures and can be identified at several levels. Low-level features are defined at a local level, such as the phonemes durations, or the pitch and intensity represented by continuous curves composed of instantaneous values that can be specified at the frame level. Those features should be generated by the control module and are the direct input of the synthesis module (as illustrated in figure 1.1). Then, upon those low-level features, higher-level ones can be built, such as vibrato, crescendo or portamento (smooth transition between 2 notes). Those high-level features span over longer time windows and are an important vector of expressivity. Depending on the chosen approach, the control module can use specific models to describe and generate the low-level features using a set of higher-level parameters (e.g. vibrato rate, transition duration, attack sharpness, ...), or directly generate the low-level features (e.g. frame-level f_0 values).
The main features that may be controlled for the synthesis, depending on the system, are: the fundamental frequency (f_0), the phonemes' durations and positions, the intensity, and some timbre-related features (related to the vocal tract characteristics or voice quality). In [Cha13], the author did a thorough musicological study of all the possible aspects related to the interpretation in singing. In [Umb+15], the author proposed a classification of these various parameters into melody-related, dynamics-related, rhythm-related, and timbre-related features. Note that some high-level features may also be considered as transverse, such as the vibrato that is mainly related to pitch, but can also have an influence on the intensity and timbre.
Regarding the control of the synthesis, we will mainly focus in this thesis on the 3 low-level features that seem really essential to singing synthesis, as being directly measurable and perceptible, and accessible to all presented signal models, that is: the f_0 variations, the temporal alignment (positions and durations) of the phonemes, and the intensity variations, also embedding the higher-level features we mentioned. To a lesser extent, we will also mention some timbre-related aspects of control. To illustrate those features, figure 2.9 shows as an example an extract

of a singing voice recording with the aligned spectrogram, midi notes, phonemes boundaries, f_0 and loudness curves. In the following subsections, we detail the characteristics of these main control parameters. Note that we don't consider here the case of physical modeling synthesis for which control parameters would be related to physical and physiological dimensions of the vocal apparatus such as sub-glottal pressure, vocal fold tenseness, or tongue position, for instance.
Other style-specific characteristics such as subtle rhythmical variations (e.g. swing or small time lags) or ornamental notes are also important features, but can be described in the symbolic domain, and should thus better be handled at the score level (e.g. by modifying the durations in a midi or musicXML file). Those aspects are thus not considered as part of the control module in the present work, although some style-related symbolic processing of the score may be later included.
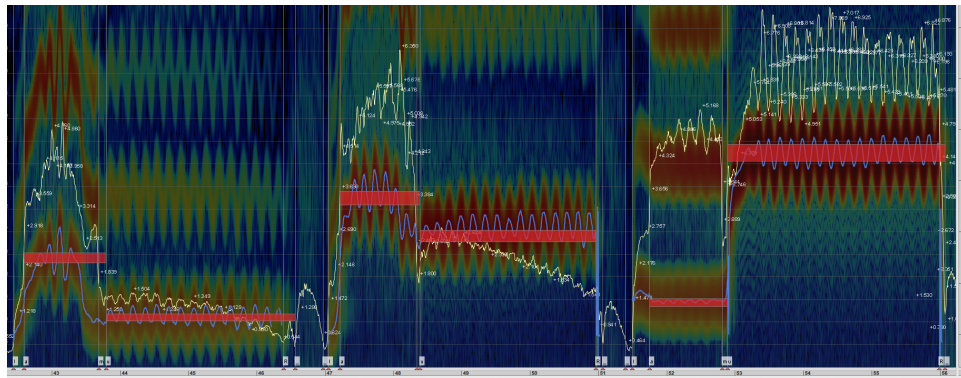
FIGURE 2.9: Extract of a singing voice recording with the aligned midi notes (red horizontal bars) and main control features: phonetic segmentation (vertical lines), f_0 curve (in blue), and loudness curve (in white)

2.6.1.1 Fundamental frequency (f_0)

Among all control parameters, fundamental frequency (also denoted as f_0, or pitch) is probably the most important one, as it is the pitch, more than the rhythmical
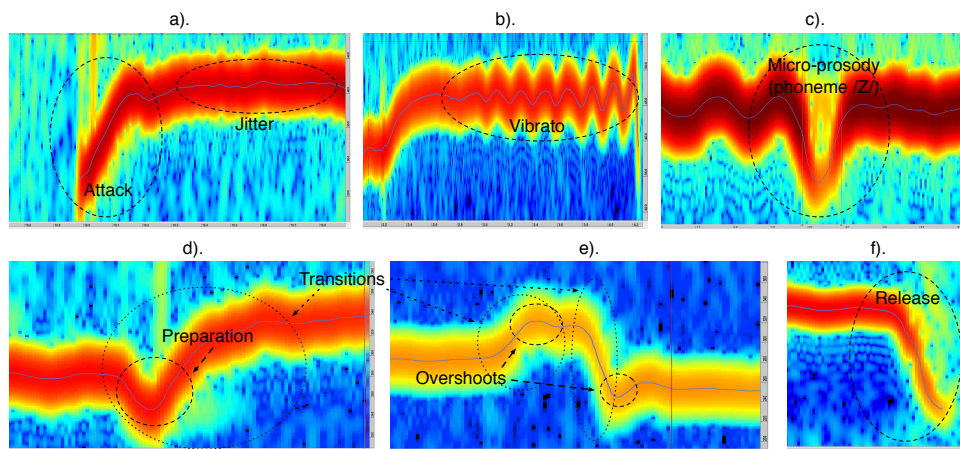
FIGURE 2.10: Example of f_0 contours with identified characteristic fluctuations.

aspect or any other, that determines the identity of a melody and allows us to recognize a song. Furthermore, beyond carrying the melody, the f_0 variations also conveys music style, personal expressivity and other characteristics specific to voice production mechanism [SG09; Cha13; Kak+09; NLM07]. The f_0 modeling is therefore critical for a natural-sounding and expressive synthesis and should thus be considered in priority.
Various approaches can be used to estimate the f_0 curve on a monophonic sound with reliable quality (e.g. [CK02; CH08; KAZ16]). Figure 2.10 shows some extracts of f_0 contours of a singing voice, where various types of fluctuations have been identified. Some of these fluctuations are due to uncontrolled behaviour related the voice mechanism and articulation, independently of the skills of the singer, and confer some naturalness to the voice. Other types of fluctuations are controlled and used as expressive means to interpret a melody in relation to singing style and aesthetic qualities of the voice, requiring a certain level of proficiency from the singer to be properly controlled.

We describe here these various fluctuations that constitute the f_0 contour of singing voices:

- **Jitter:** Even when trying to sing a very stable sustained note, it is not possible for a human to keep a perfectly constant pitch when singing. Jitter is an involuntary random perturbation of the f_0 during phonation. In [Sta11], the author talks about "pitch drift" as a low-frequency perturbation occurring below the vibrato frequency (5Hz), whereas in [Sai+07], authors refer to "fine fluctuation" as an irregular frequency fluctuation at rates higher than 10Hz. In this thesis, we will refer to jitter as any kind of random fluctuations that are not related to expressive intentions of the singer or to articulation, thus encompassing the pitch drift and fine fluctuations evoked in [Sta11] and [Sai+07]. Some jitter can be observed on figure 2.10 a).

- **Micro-prosody:** In speech and singing, the articulation (succession of phonemes pronounced) does not only affects the timbre but also the pitch and intensity. Micro-prosody thus denotes phoneme-dependant variations that affect the f_0 and intensity contours, and has been evoked in many studies [Bon08a; STK10; Umb+15]. While certain phonemes may not induce much variations in their pitch contours, this effect is particularly important in the case of voice fricatives (phonemes /v/, /z/, and /Z/ in SAMPA notation), which systematically induce a pitch valley, as can be seen in figure 2.10 c) for phoneme /Z/.

- **Vibrato:** Vibrato is defined in [Sun90] as a periodic, rather sinusoidal, modulation of the f_0. Vibrato is characterized by a rate (or frequency) parameter that tends to range mainly from 5 to 7.5Hz, depending on the singer, and a depth parameter usually ranging from ± 0.5 up to ± 2 semitones. Vibrato in singing voice has been widely studied [Pra94; MB90; BS02; DAl94], mostly from an analysis or perceptual point of view. Vibrato is assumed to be a natural behaviour related to the voice mechanism [Sun90], that trained singers may be able to produce rather unconsciously. Singers can hardly control the rate of the vibrato, which tends to be rather stable for a particular singer, but they can better control its depth in order to use it more or less prominently, for expressive purposes. Vibrato is present in many singing styles, and is especially important in lyrical (western operatic) singing. Some phenomenons

such as an increase of the vibrato rate on the last cycles [Pra94] have been reported. However, [MB90] suggests that a good vibrato is nearly sinusoidal and that changes in its shape along time is not perceived by listeners. While some authors have seeked to precisely characterize the shape, rate, and amplitude of each vibrato cycle for synthesis [STK10; IIO14a], the necessity of such a precise description of the vibrato for synthesis purpose, from a perceptual point of view, has not been attested.
As stated in [RPB84], the vibrato is especially important for recognizing the identity of a singer, not only because of the specific vibrato properties of that singer (and particularly its frequency), but also because the frequency modulation induced sweeps across the formants along time, which thus better reveals the voice timbre than with a flat pitch. This is especially true for high-pitched voices for which the spectral envelope is poorly sampled by the harmonics.

- **Preparations and overshoots:** During continuous (voiced) transitions between 2 notes, the f_0 contour exhibits some fluctuations that have been denoted in [Sai+07] as "preparation" and "overshoot". An overshoot is an inflection exceeding the target note's pitch after a note change, at the end of a transition. In [SUA05], the authors stated that overshoots are the type of inflections that affect the most the perceived naturalness of synthesized voices. A preparation similarly denotes a deflection in the opposite direction to the note change, at the beginning of a transition. Examples of both can be observed in figure 2.10 d) and e).

- **Attacks and releases:** Regarding the f_0, the term "attack" designates a rise of the f_0 contour, starting below the target note, at the begining of a sentence after a silence, as shown on figure 2.10 a). Contrarily, a release designates a fall of the contour at the end of a sentence, just before a silence, as shown on figure 2.10 f).

All those components affect the naturalness and/or expressivity of the voice and should thus be appropriately modeled. Many different approaches to generate f_0 curves for singing voice synthesis have been developed. Some approaches are based on parametric models and rules, while other mainly rely on the use of a database of singing recordings. Some parametric models specific to f_0 modeling will be briefly introduced here, while the main approaches to expression control, including f_0 modeling, will be developed in section 2.6.2.
In [SUA02; SUA05; Sai+07], some expressive f_0 variations like preparation, overshoot, or vibrato are generated using the transfer function of a 2^{nd} order linear system, parametrized by a damping coefficient, a gain factor, and a natural resonant frequency. The curve is obtained by computing the response of this system to the melody component described as a step function of the notes' pitches. 3 functions with different parameters are used for the preparation, overshoot, and vibrato fluctuations. Fine-fluctuations (or jitter) are added using filtered white noise. In [Ohi+10], the autors proposed an approach to automatically estimate the parameters of such a system.
Inspired from the Fusijaki model for speech [FH84], [Ohi+12] proposed another approach that uses the same idea of exploiting 2^{nd} order models, but this time separating the input in notes and expressions commands as 2 separate step functions, one for transitions and overshoots, and the other for vibrato and portamento.
In [Dev+11], the Discrete Cosine Transform (DCT) is used to characterize singing

voice f_0 trajectories. However, although it is mentioned as a perspective, the study doesn't propose a model for generating the f_0 from a score.
In [Bon08a], the author proposed to generate a baseline f_0 curve based on heuristic rules, according to observation of recordings. This approach consists in smoothly interpolating (using linear plus squared sinus segments) a set of points empirically obtained from normal distributions, corresponding to predominant curve tendencies (depending on the notes' durations). In addition, some parametrized templates are used for generating expression (e.g. for vibrato). The author of this work also introduced the notions of "portamento" defined as reaching a note pitch before the note onset, and "scoop", defined as beginning the note transition after the actual note onset.
This last approach may be considered to belong to the category of rule-based approaches. The main other techniques employed for generating f_0 curves are based on HMMs and units selection, that will be explained in section 2.6.2.

In this thesis, we propose an new model for the generation of the f_0 curve from the score and lyrics, which has been the subject of a publication [ADR15]. This model will be throughly presented in chapter 4.

2.6.1.2 Timing

While melody is probably the first element perceived and recognized in a song, rhythm is another essential aspect that is directly represented in the score. It determines the start and end time of each note of the melody. We denote here by "timing" all control features that relate to the rhythmical dimension of a singing performance. As said before, whereas rhythm is a rather simple and explicit attribute for some instruments like piano, several aspects are to be considered for the singing voice, due to the presence of lyrics: the temporal positions of the notes, related to the notes' nominal durations obtained from the score (and to potential rhythmical deviations); the timing of phonemes; and the relation between both, that we will refer to as "temporal alignment".
In real singing performances, the tempo is generally not perfectly constant throughout the interpretation of a score, and some timing deviations from the notes' positions and durations given by the score can be observed. As said previously, works have been conducted on expressive performance rendering, that work in the symbolic domain (e.g. by modifying midi files) to apply rhythmical variations [WG04; Fri+00] to the initial score. In this thesis, we don't consider this aspect and will only focus on the phonemes' durations and their positions in relation to the notes boundaries, that are assumed to be precisely defined in the input score, without further modification.

In order to be able to sing a melody, any note present in the score has to be associated with only one vowel. Each vowel can be part of a syllable comprising one or several consonants before the vowel, but it is not possible to sing a note only on a consonant. Sometimes, a single vowel could also be sung on several succeeding notes, which is called a "melisma". On a written score, each syllable is usually associated to a note, and one could intuitively think that the start of syllables should be aligned with the notes' onsets. However, it is a common agreement, reported in many studies, that notes' onsets should match the onset of the associated vowels (and not that of the syllable), and this rule has thus already been implemented into many singing synthesizers [Sun06; Mac+97b; Une02;

Bon+01a; KO07; Bon08a]. In case the syllable contains one or several consonants, those consonants should thus be pronounced before the note onset, during the time frame of the previous note.

Besides this temporal alignment between vowels and notes' onsets, phonemes durations should be properly generated for synthesis. Phonemes' durations may depend on many factors, from phoneme type or identity to inter-individual variations, rhythm and tempo, or expressive intentions, as some syllables may be accentuated by purposely extending the consonants durations (as it is sometimes also the case for prosody in speaking voice [Bel09; Obi11]). But few studies explicitly model the phonemes durations for singing synthesis.
In [Sai+04; Sai+07] a rule-based approach is used for controlling phonemes' durations according to the note's duration, for speech-to-singing conversion. In this work, each "mora" (the basic phonetic unit of Japanese language) is decomposed into 3 parts: a consonant part, a boundary (co-articulation) part, and a vowel part. Then, the consonant part is stretched using a fixed factor depending on the type of consonant (fricative, plosive, nasal, or semi-vowel), the boundary part is not modified, and the vowel part is stretched so that the total length matches the note's duration. The fixed stretching ratios for the consonants have been determined empirically by comparing speaking and singing voices. This approach assumes that each consonant has a "natural" duration, different in speaking and singing, independently of the note's duration, but no adaptation is proposed for higher tempo (and thus shorter notes), in which case this simple strategy would not be very well suited. A strategy would thus be necessary to adapt phonemes' durations in the case of fast singing. However, this highlights the importance of the consonants' type for duration modeling. This example also shows the importance of the language as a potential influential factor on phonemes' durations and positions. In this thesis, we will only consider the case of the French language.
HMM-based approaches like [Sai+06] implicitly model phonemes' durations based on duration modeling of context-dependent states, using decision-tree based context-clustering. In [Sai+06], the authors further introduced a "time-lag" model to infer the temporal alignment between the phonemes and the notes positions (instead of strictly aligning vowels to notes onsets). These time-lags are estimated simultaneously to the states durations so as to maximize their joint probability.
In [Une02], an approach is presented for computing the phonemes' durations and alignment with notes, according to the notes' durations, that differentiates between the various type of consonants for applying a compression rate in short notes, based on the minimal and standard durations of each phoneme (computed from recordings at fast and moderate tempi).
Our own approach for dealing with the timing and durations of phonemes will be presented in chapter 4.

2.6.1.3 Intensity

A 3rd important parameter of singing voice is intensity. The intensity is a measure of energy in the acoustic signal, related to the perceived loudness of the voice, that can be measured at a frame level to obtain a continuous curve. The temporal variations of intensity are often referred to as "dynamics".
Several measures can be used to estimate the intensity contour of a voice signal. A first possibility is to use the simple Root Mean Square (RMS) [DD98]. However, this measure is directly linked to the amplitude (or energy) of the signal's

waveform, but doesn't take into account the specificities of human perception of sounds, as for instance the sensibility to certain frequency regions, or masking effects. As an alternative, loudness models can be used for better measuring the perceived intensity of a sound, taking into account the various psycho-acoustic effects related to hearing. But, due to the complexity of the hearing system, depending on the nature of the sound, no single model is perfect and several more or less complex models can be used. [FPR11] gives a good and recent review of the main existing loudness models. However, the hearing system is less sensitive to loudness variations than to pitch fluctuations, and the precision of this measure in thus not as critical as for the f_0.

The intensity fluctuations along time are affected (more than pitch) by micro-prosody, the loudness level being determined by the energy repartition along the frequency axis, that depends on the phonetic content of the pronounced lyrics.
Intensity may also be correlated, to some extent, to other parameters like pitch, as it is not possible for a singer to sing very loud with a low pitch, and the perceived intensity tends to increase with the f_0, as reported in [Gra+88].

Tremolo is the intensity-related counterpart of vibrato, and can be described as a periodic fluctuation of the intensity during sustained vowels. As explained in [Sun90], tremolo is partly directly induced by the vibrato oscillation, as when harmonics move in frequency, changing their distance to a formant's centre frequency, their amplitude vary according to the amplitude and bandwidth of the formant, thus inducing a periodical fluctuation of the overall intensity of the signal, as has been studied in [MB90]. Depending on the pitch, and thus the frequency position of the strongest harmonics relatively to formants' frequencies, the tremolo may be in phase or in phase opposition with the vibrato, or out of phase, exhibiting modulations at twice the vibrato frequency (if the harmonics sweep back and forth between the left and right sides of a formant). But some singers may also intentionally use tremolo as an expressive feature in itself, independently of vibrato, by emphasizing this amplitude variation on purpose. [Cha13] evokes for instance the case of the French singer Véronique Sanson, who uses this "intensity vibrato" particularly intensively. However, from a perceptual point of view, tremolo remains secondary compared to the f_0 modulation of vibrato.

Besides micro-prosody and tremolo, some variations of intensity can be used to convey expressive intentions and add some structure to a musical piece at different levels. These variations can be related to some nuances written in the score, or freely performed by the singer in its own interpretation, and shape sequences of notes or single notes at a finer level. For instance, some notes or words can be accentuated by emphasizing the difference of intensity relatively to the surrounding ones, and crescendi or decrescendi can be used for shaping a sustained note or a group of notes to add some phrasing structure. Such intentional variations are related to the vocal effort produced by a singer to project his voice and thus are correlated to timbral attributes of the voice source such as spectral slope.
An approach to generate expressive variations of intensity in synthesis will be presented in chapter 4.

2.6.1.4 Timbre-related features

Although timbral parameters are tightly related to the employed signal modeling approach and can not always be easily modified, we give here, for the sake of completeness, some insight on how such features may be controlled, either explicitly or implicitly.

The timbre mainly depends on the shape of the vocal tract and on the mechanical characteristics of the vocal folds or other vibratory mechanism of the vocal apparatus which affect the voice source signal. As seen in 2.5, timbral features cover varied aspects of singing, and their control can thus imply very different requirements, depending on the feature. The timbral features related to phoneme identity (i.e. formants positions) are implicitely defined and modeled based on the known phonemes sequence, and thus don't require additional control. Some other timbral features are directly related to parameters like pitch or intensity. In [Feu+17], the glottal formant's frequency and spectral tilt of the source model and the 1st formant's frequency are directly related to the vocal effort, and thus to the intensity, based on predefined rules. In that case, no explicit control of source and formants' parameters are thus required. Similarly, formant tuning strategies can be used to automatically adapt formant's frequencies according to pitch [HSW11]. The vocal register (i.e. chest or falsetto), dependant of the pitch, is related to the source parameters, and may be controlled either explicitly or relatively to the pitch based on a predefined threshold.
But the voice source and vocal tract's characteristics are also used differently among singing styles [TS01; BK06]. A particular singing style could thus be chosen, which may affect various timbral features, as it is the case for instance in the Cantor Digitalis software [7], where different types of voices (e.g "Bulgarian-style singer") can be chosen. As explained in section 2.5, male operatic singing is characterized by the presence of a singer's formant. Some parameters could be used to give this operatic characteristic to a voice by applying such an effect.
Finally, voice qualities such as roughness, tenseness, or breathiness could be expressively controlled, by varying the degree of those effects along time. In [LB04], the authors proposed an automatic control of the growl effect, to determine where and how it should be applied, based on the derivative of the f_0 and energy contours.

2.6.2 Main approaches to expression control

Depending on the given inputs, different types of systems can be identified, regarding the control of the synthesis.
Real-time synthesis systems like [Feu+17; Coo05] make use of direct explicit controls of the various parameters given by a musician, using a dedicated Human-Computer Interface (e.g. a graphical tablet).
Another kind of systems, called "performance-driven systems", makes use of the real performance of a human singer to extract expressive parameters and use it to synthesize a similar performance with a different voice [JBB06; NG09], which requires to have a recording of the target song. A particular interest of such system is to compare the quality of the synthesizer to that of a real singing voice, for instance to evaluate the quality of the signal modeling and transformations

[7] https://cantordigitalis.limsi.fr/
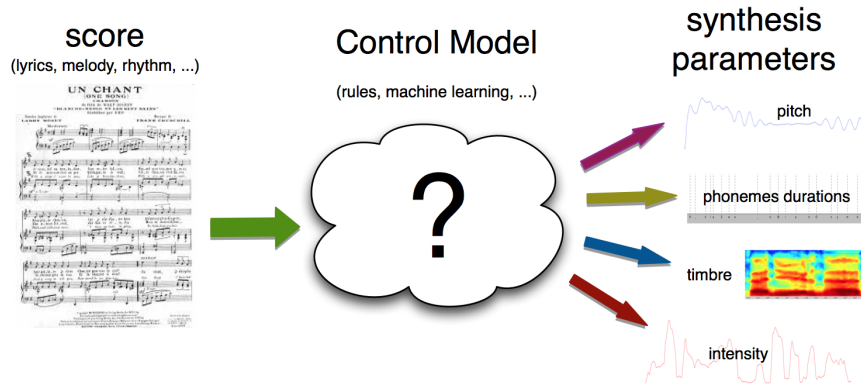
FIGURE 2.11: Illustration of the control module

independently of the control. When the parameters are directly copied from the real singing voice to the synthesizer's input, we can also talk about "copy synthesis".
The 3rd type of systems takes as input a pre-defined score with associated lyrics, from which all the necessary control parameters are generated, as illustrated in figure 2.11. In the context of this thesis, we only consider this last type of systems for producing offline synthesis from a score and lyrics.
In that case, features can be generated either completely automatically using only the symbolic informations from the score, or possibly partly determined by the user interactively using an appropriate user interface. Different degrees of interaction may be allowed to the user to manually tune a wide set of control parameters. In [Bon08a], the user can select the voice quality parameters, timing (notes onsets and durations), dynamics, musical articulation (e.g. soft or sharp attacks and releases, or transition's type), and other expressive features (vibrato and tremolo). The Vocaloid software[8] offers a graphical interface from which users can manually tune features like pitch deviations and growl, using general settings and automations, to obtain a satisfying result. But the manual tuning of all control parameters is a difficult and time-consuming task to obtain realistic performances. While it is desirable for the user to have a direct control over the expressive parameters, synthesis systems should thus be able to produce convincing results directly from a score.

Because of the versatility of singing voice, one can however hardly expect to come up with universal rules and models that are well-suited for any singing style, which should be taken into account when building a singing synthesis system. Moreover, many different interpretations of a same score may be acceptable. Assuming that different singers have more or less access to similar expressive features, the singing style of a singer results from the use of certain configurations of expressive control parameters occurring in certain musical contexts. Under these considerations, the problematic of singing style modeling can thus not be considered separately from the general problem of expression control. However, the precise definition of a singing style in the framework of singing synthesis has not been well established, and will thus be subject to further discussion in chapter 5.

[Umb+15] and [Umb15] give a very recent and thorough review of expression

[8] https://www.vocaloid.com/en/

control for singing voice synthesis, which provides a very beneficial source of informations and references. Along with the article, the authors have gathered some sounds produced by the various systems that can be found at the url[9].
3 main approaches for controlling expression from a score and lyrics can be identified: rule-based methods; statistical modeling (HMM); and units selection. In [Umb15], hybrid approaches have also been recently proposed, that try to take the best of both HMM-based and units selection techniques. Additionally, another recent approach based on the selection of parametric templates has been presented in [IIO14b]. We present below those different approaches to expression control.

2.6.2.1 Rule-based approaches

In rule-based approaches, rules are empirically defined and "hard-coded" into the synthesizer. Such systems benefit from expert musical knowledge and can be progressively improved, while generating more knowledge, using an analysis-by-synthesis procedure. This procedure is typically based on trials and errors to identify acoustic features that are perceptually relevant. A first tentative rule can be tried out and the result of the synthesis is (informally) assessed. Then, depending on the result, the rule can be changed or refined and assessed again, iteratively. The defined rules can be combined to model different musical styles.

An example of such system is the KTH rule system for music performance [FBS09] and for singing synthesis [Ber96; Sun07; Sun06]. The KTH system consists of a large set of performance rules that predict the timing, dynamics, pitch, and timbral features. Most of the rules look only at very local contexts (e.g. simple ratios of durations or differences of pitch between successive notes, ascending lines, etc...) and affect individual notes, but some higher-level rules may also refer to an entire musical phrase. For instance, in [Fri91], the defined rule "DDC 2A" for music performance applies specific amplitude envelopes to accentuate the *"first of several equally short notes followed by a longer tone"*. Rules described in [FBS09] have been implemented in the Director Musices software [Fri+00]. Some of these rules are described in [BF99] to apply various emotions to music performances. For singing voice, a selection of the KTH rules has been applied to the Vocaloid singing synthesizer in [Alo04]. [Fon01] also implemented some rules adapted to singing voice, at both the note and phrase levels, based on research from [Fri91] and [Ber96]. In [Bon08a], the author used some kind of heuristic rules obtained by observing recorded singing voice performances to control pitch fluctuations by smoothly interpolating a set of points obtained from normal distributions.

An advantage of this approach is that the implementation is relatively straightforward and fully deterministic, although some random variations can possibly be introduced so that each synthesis of the same score is slightly different. Another advantage is that this approach can be used without a database of recording, using only the input score and implemented rules, while the approaches presented in the next sections require large sets of annotated data.
The main drawback of this approach is its lack of flexibility, as a thorough musicological study is necessary to define each rule, and representing a new singing style with its specific rules is thus a long and fastidious task. The rules are

[9] http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis?p=Signal_Processing_Magazine_2015

thus also likely to be inferred based only on a few observations that do not fully represent a given style.

Opposed to this qualitative knowledge-based approach, statistical methods try to automatically infer the control parameters from data, using machine learning.

2.6.2.2 Statistical approaches

Although statistical approaches are very different from rule-based approaches, they still shares some similarities, in the sense that they choose the parameters to be applied according to contexts, which can also be interpreted as rules, except that those rules are inferred from automatic large-scale data analysis instead of being hard-coded and can thus easily be modified only by tweaking the data, without having to change the code behind. In [WG04], authors stated that machine-learning techniques allowed to automatically discover rules bearing a strong resemblance with that of the KTH model.

The main type of statistical approach are HMM-based approaches, already evoked in section 2.3.4 for generating the voice signal. Beyond generating the voice spectrum, those systems are also able to generate the control parameters, as explained for instance in [Sai+06; STK10; Our+10; Lee+12; LDL12]. Although those systems implement different variants and improvements, they all basically work the same way. A particular advantage of this approach is that it can jointly model the various features.
These systems operate in two phases: training and synthesis. In the training part, acoustic features such as f_0, vibrato-specific parameters, intensity, and MFCCs are estimated from a database of annotated recordings, and associated with contextual labels obtained from the annotations. Depending on the system, contextual labels can relate to different levels, from the phonemes and notes to the entire musical phrase or the whole song. A few examples of contextual labels are: syllable identity; identity of previous, current and next phonemes; pitch of the previous, current and next notes; notes intervals; notes durations; positions of notes in a musical bar; tempo; ... Then, those contexts are clustered using decision trees and HMMs are built from statistics computed on each cluster, that relate how the estimated expression features behave according to the contexts.
During the synthesis part, contextual labels are derived from a target score (notes sequence and lyrics), and all parameters (state durations, f_0, vibrato, and MFCCs observations) can be directly generated from the HMM, based on those contexts. In some systems, the dynamics may be implicitly modeled in the MFCCs along with timbre. Phonemes durations are usually implicitly modeled from the HMM states durations. In [Sai+06], time-lag models are also used to infer phonetic timing. Note that if the timbre (MFCCs) is not modeled, the generated control parameters can still be used as input to another kind of synthesizer (e.g. concatenative synthesis), as is done in [STK10] and [Umb15].

In [Umb15], 2 variants of HMM-based systems for expression control are presented. The first one models the pitch and dynamic contours on a note basis (using 5 states per note). The second one proposes to rather model sequences of transitions and sustains segments (attacks and releases being considered as transition segments). In this case, one model is built for the sustain segments, and 5 models for the transition (attack, release, ascending transition, descending

transition, and transition smaller to a semitone). In [STK10], the notes are supposed to be divided into up to 3 regions ("beginning", "sustained", and "ending") with different types of behaviour, and a label representing this segmentation is predicted for each note, that is then used as a contextual factor for generating the f_0 and power. Another difference between systems is that in some cases, the absolute pitch values are used in the modeling, while in others, pitch values are relative to a nominal pitch contour generated from the melody, so that all pitches don't need to be covered in the database.

Conversely to rule-based approaches, HMM-based approaches can model new singing styles by choosing an appropriate learning database, without requiring specific knowledge. The database used for the training should thus be built to target a specific singing style. For instance, in [Umb15], a corpus of 17 jazz standards recorded by 1 singer is used as an expression database to capture the specific style of this singer. In [STK10], a database of Japanese children's songs is used to learn the f_0 and dynamics parameters.
The HMM-based approach is also quite flexible, as new voice characteristics can be easily generated by modifying the HMM parameters using model adaptation [Tam+01a] or interpolation [Yos+00; TD12] techniques, thus allowing a global high-level control of the synthesis. For instance, it is proposed in [Nos+15] to continuously control the degree of different singing styles (tested on age-related styles labeled "child-like" or "adult-like" singing). In the proposed technique, singing styles and their intensities are represented by low-dimensional "style vectors", assuming that the mean parameters of the modeled acoustic features are given as multiple regressions of those style vectors. In the synthesis process, it is then possible to weaken or emphasize the weightings of singing styles by modifying the style vector.
Another strength of HMM-based system is the rich contextual description used in the contexts clustering, where the contextual factors are automatically selected based on their influence on the measured acoustic parameters. In [Nos+15], many contextual factors have been used for the training, about the identity of the current and surrounding phonemes, the absolute and relative pitch of current and surrounding notes, as well as their durations, and the position of the notes in the bar.

But a drawback of such approach is also that besides a possible global control of the singing style, it does not provide any local control of the expressivity, at the note level, to the user.
Another drawback of HMM-based systems is that an important quantity of properly annotated data is often necessary to cover a sufficiently large set of contexts with enough redundancy to allow a robust learning, for a particular singer, singing style, and language. While the approach presented in [Nos+15] seems to provide good results in the presented evaluation, 25 songs were used for the training, which is a rather large amount of data.
In other works where the voice spectrum is also jointly modeled, even more data have been used. In [Sai+06], 60 songs (72 minutes) of a male voice singing 60 Japanese children's songs were used for the training in order to mimic the voice quality and singing style. A similar system described in [Our+10] used 70 minutes of a female voice singing 70 Japanese children's songs as a training database. The advantage of using more data is a better coverage when using many contextual factors, but the consistency of the expressive features using so much recordings may be questionable. Besides the work load to constitute such large databases, a

possible drawback is thus an oversmoothing of the generated data.
In [STK10], only about 5 minutes of recordings were used, but only the f_0 and power (intensity) were modeled, and only the MIDI note number (pitch in semitone) and duration of notes were considered as contextual factors.

Another new type of statistical method are neural-network-based approaches, as already evoked in section 2.3.5. An example is [BB17], in which authors obtained better scores for the neural-network-based approach compared to an HMM-based approach in a preference test, using rather small databases of 16 to 35 minutes.

2.6.2.3 Unit selection-based approaches

Concatenative units selection-based systems where presented in section 2.3.3 for generating the voice signal. These systems also inspired a similar approach for generating the expression contours using units selection, where the selected units are pitch and intensity contours rather than sound samples [UBB13a; Umb+15; Umb15]. For this purpose, a second database specific to expression must be provided. With this approach, the knowledge and skills of the singer are implicitly contained in the recordings.

Similarly to what was explained in section 2.3.3, the main idea of units selection is to select small segments of f_0 and intensity contours from the expression database, and then concatenate and transform them according to the target score. In units selection-based approaches, these segments are chosen, based on contextual factors, using cost functions. As in section 2.3.3, both a target (or transformation) cost and a concatenation costs are used. The target cost measures how much a unit has to be transformed to match the target score (notes pitches and durations). The concatenation cost measures the perceptual consequences of joining 2 units. These cost functions are then weighted and summed together, and the sequence of units yielding the lowest overall cost is then selected using the Viterbi algorithm.
The main issues for this technique are thus: to design an appropriate database that can represent well all expressive variations of a particular singer or singing style, covering a large set of contexts, similarly to HMM-based systems; and to design appropriate cost functions according to the relative importance of the various contextual and perceptual factors.

In [UBB13a; Umb15], authors used a database sung using only vowels in order to avoid the effects of micro-prosody, that are not related to expression, when extracting pitch and dynamics. The songs from the database were labelled in a semi-automatic procedure, with the timing and pitches of notes. The annotation of the vibrato segments was manually corrected and the vibrato parameters (depth and rate) were extracted, in addition to the raw pitch contour. From this database, sequences of 3 consecutive notes or silences are use as units to represent the local context.
In [UBB13a; Umb15], the transformation cost is expressed as the mean of 2 sub-cost functions representing the amount of pitch shift and the amount of time-streching required to match the target values. The concatenation cost measures how well 2 units can overlap, based on the times of the transition parts that are cross-faded (this cost is 0 in the case of consecutive units). A continuity cost is

also used in order to further favour the selection of long sequences of consecutive notes from the same song in the database. This allows to obtain a result as close as possible to the real expression of the singer. Once the tri-notes units have been chosen, they are transformed, concatenated, and overlap-added to generate the final curve.
For the computation of the concatenation cost, the annotation of the transitions' start and end times are also required, as well as for time-scaling the units, so that only the sustain parts are stretched, without altering the notes' transitions. The vibrato is modeled parametrically, separately from the baseline pitch contour so that the notes can be properly overlapped and stretched without altering the vibrato.

An advantage of this approach compared to statistical ones is that it directly applies the expression features of the singers with all their fine details, without suffering from over-smoothing. As their is no statistical modeling, smaller databases can be used. The experiment presented in [UBB13a] used only 6 minutes of recording in soul/pop style.
A disadvantage of this method is however that the annotation work can be fastidious, especially for annotating the vibrato and transition parts, that are done partly manually. Another drawback is that the control features are not parametrized (except for the vibrato), and can thus not be easily modified (besides manually redrawing the curve). Finally, the use of an empirically-defined cost function doesn't allow to use a rich context description as is done in HMM-based approaches using decision tree-based contexts clustering.

2.6.2.4 Hybrid approaches

We presented so far the 3 main existing approaches to generate expression features in singing synthesis systems. As we have seen, each of those approaches have advantages and drawbacks. In [UBB13a], it is suggested that *"rule-based approaches would benefit from machine-learning techniques that learn rules from singing voice recordings to characterize a particular singer and explore how these are combined"*, and that *"the combination of existing approaches [has] great potential"*. In [Umb15], the author thus proposed an hybrid approach that aims at combining several methods in order to overcome some limitations while keeping the best of each method.
The idea behind the proposed hybrid system is to first use a HMM-based approach to generate initial contours that are then used to enrich a cost function to guide a units selection system. During the computation of the cost functions, the candidate units from the expression database are compared to the statistically generated baseline pitch (without the vibrato). Compared to the previous units selection approach of [UBB13a], a new additional cost function is thus proposed, based on Dynamic Time Warping (DTW) to measure the distance between the units and the target contour generated by the HMM-based system. The main advantages of such an approach is the possibility to use the extended contextual information from HMM systems (compared to the unit selection which can use only a limited set of contexts in the definition of the cost functions), while reproducing the fine details of the original contours with units selection.

2.6.2.5 Parametrized expression templates selection

A last recent approach to expression control, focusing on f_0 contours, has been presented in [IIO14b]. In this approach, some expressions such as vibrato, glissando (attacks and releases) and kobushi (ornament specific to Japanese singing) are first segmented on commercial polyphonic recordings of famous Japanese singers and parametrized using specific models, as explained in [IIO14a; IIO14b]. Once each vocal expression has been parametrized, those parametric templates are extracted, along with the local notes contexts (note pitch, duration, and labels for notes at the beginning and end of a phrase), to form a vocal expression library. For synthesis, some of those expressions can then be chosen for each note, according to the difference between the original and target notes' contexts (similarly to the cost functions used in unit-selection approaches), to form the pitch contour.
[Bon08a] shared a similar idea of using (partly) parametrized attacks, releases, vibrato, and transitions templates chosen from a performance database to transform sound units in a concatenative synthesizer. But not much information is given about how the templates are selected.
The main interest of this approach is that it allows to characterize quantitatively, using a restricted set of parameters, the expressive variations of the control parameters.
But, similarly to the unit selection-based approach, the use of a cost function to measure the distance between the source and target contexts allows to use only a restricted and fixed set of contextual informations, which can't represent the possible variable importance of some contextual factors from one style to another.

2.7 Conclusion

The present chapter aimed at:

- First, to give a thorough overview of the various aspects implied in singing voice synthesis;

- Then, for each aspect studied, to highlight the advantages and drawbacks of each potential technique in order to choose the most appropriate ones in our research and try to find appropriate solutions to overcome their limitations;

- Finally, to summarize the current state of the research in this field, and identify the next necessary steps towards building better quality systems for synthesizing more natural and expressive singing voices in a variety of styles.

For this purpose, we reviewed in this chapter the essential theoretical basis and the main state-of-the-art techniques involved in the various aspects of singing voice synthesis and processing, along with their specificities and limits.
First, the physiological basis of voice production has been explained with the implication of the various components.
From the signal point of view, the well-known source-filter model of voice, inspired from the vocal production system, has then been explained, along with approaches to modeling its source and filter components. In particular, the LF

glottal source model has been introduced, and the main techniques for spectral envelope estimation have been explained.
Based on this knowledge, the various possible approaches for synthesizing a singing voice have then been reviewed, including mainly formant synthesis, physical modeling, concatenative synthesis, and HMM-based synthesis, and the advantages and limits of each one have been highlighted.
As several signal modeling and manipulation approaches can be used in different systems, the main signal models and approaches have been presented, covering both time-domain and frequency-domain techniques, as well as general and voice-specific models that may be used for our purpose. In particular, the phase vocoder and superVP software have been introduced, as well as the SVLN and PSY approaches, which will constitute the main background for the development of our synthesis system that will be presented in the next chapter.
Then, some expressive transformations of the voice, necessary for improving naturalness during pitch and intensity modifications and for modeling a wider variety of timbre and voice qualities such as vocal roughness, were presented.
Finally, as we aim at synthesizing singing from a score and lyrics, a last part was related to the automatic generation of all the necessary control parameters, mainly including the f_0, intensity, and phonemes durations, and some considerations about singing styles where evoked.

From this review, we can now distinguish 3 main subjects to be further developed:

- Synthesis techniques and signal models
- Advanced expressive timbre transformations
- Expression control and singing style modeling

The various contributions of this thesis regarding each of these 3 subjects will know be developed in the following chapters, starting with the description of a concatenative synthesis system that we implemented.

Chapter 3

ISiS: a concatenative singing synthesizer

3.1 Introduction

We reviewed in section 2.3 the various existing methods for synthesizing a singing voice. Formants synthesis is especially efficient for real-time synthesis and to better understand vocal features' behaviours using analysis-by-synthesis, as it allows a direct control over formants and sources parameters. Physical modeling synthesis is interesting for investigating links between physiological and acoustical aspects, but remains complex for obtaining a high quality, and not very intuitive from the control point of view. The currently most popular techniques for synthesizing a singing voice from a score and a text are concatenative and HMM-based systems. Concatenation-based methods are well-known for generating high-quality speech and have for some years been widely used for singing voice synthesis [KO07; BL03; Mac+97b]. While HMM-based systems like [Nak+14] may be more flexible in terms of speaker identity or singing style (e.g. using model adaptation techniques [Shi+14]), their quality is still limited by the currently used vocoding techniques and oversmoothing problems. Conversely, concatenative systems, with a minimum of transformations to keep the signal close to the original voice, lead to high-quality synthesis.
In the framework of this thesis, we aim at building a high-quality singing voice synthesis system that should be able to produce a natural timbre, ideally indistinguishable from that of a real singer. For this reason, the approach we chose for this purpose is concatenative synthesis. As a first step in this thesis work, we built a fully-functional singing synthesizer based on this technique, called ISiS (for ***I****rcam's* ***Si****nging* ***S****ynthesizer*), which we introduce in the present section.

As explained in section 2.3.3, concatenative synthesis requires the use of a database from which some sound segments (or units) are first selected and then concatenated and transformed to generate the output voice. The main transformations to be performed to match a target score with a specific melody and rhythm are transposition (or pitch-shifting) and time-stretching. The concatenation process also requires specific treatments in order to smooth out the audible discontinuities between the selected units and produce a natural-sounding continuous flow.

Figure 1.1 showed the basic building blocks of a singing voice synthesis system. Figure 3.1 further details the steps performed by the synthesis system in the case of concatenative synthesis. In this case, a sequence of units is first formed according to the input lyrics, which are then selected from a pre-annotated database. The synthesis engine is then in charge of handling both the concatenation
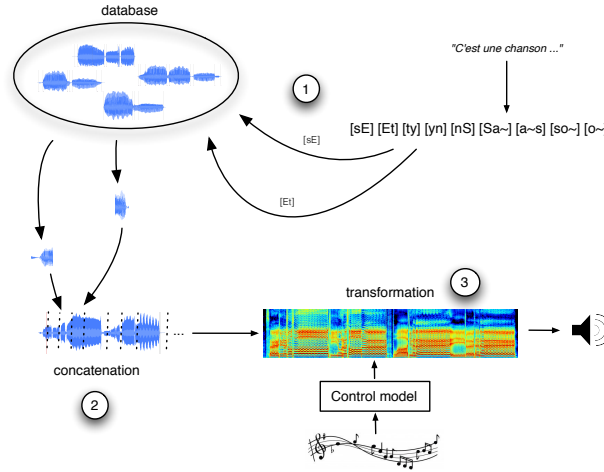
FIGURE 3.1: Overview of a concatenation-based singing voice synthesis system: First, units are selected from a database (1) to be concatenated (2) and further transformed match the target control parameters (3)

and the transformation steps. It takes as input all control parameters (mainly pitch and phonemes durations) from the control module, along with the selected units that are concatenated and transformed to smooth out discontinuities at junctions and match the target control parameters.
We conceived this system in a modular fashion, such that the unit selection process, the synthesis engine, and the control module are independent from each other, and can be easily replaced by a new one to test different approaches. In particular, several synthesis engines using different signal modeling techniques can thus be used with the same units sequence and control parameters.

This chapter will give a review of the various components of the synthesis system we developed, except for the control module which will be the subject of the next chapter of this manuscript. First, we will present the database used by our system, along with the necessary annotations. Then, we will explain the unit selection process required to choose the units to be concatenated from the database. Finally, we will present 2 different synthesis engines that have been integrated in our system, and some specific treatments required to avoid discontinuities due to the concatenation process.

3.2 Databases

Besides the signal modeling technique, the quality of a synthesized voice also greatly depends on the quality of the database used, which should thus be recorded and prepared carefully. Several databases have been built to be used by our synthesis system. We review in this section the various specifications and constraints related to building those databases, and explain our strategies for the recording and annotation processes.

3.2.1 Database construction

3.2.1.1 Specifications

In speech synthesis systems, rather large databases are typically used to cover a wide variety of prosodic contexts, from which units are selected with possibly variable sizes, with a minimum of transformation to be applied [Bla02]. However, compared to speech, the variations in singing, in terms of pitch, loudness, and timbre, cover a much wider range of possibilities. It is thus not possible to capture all the combinations of pitch, loudness and timbre with a reasonable quantity of recordings, and we have to rely on transformation techniques in order to cover a range as wide as possible with a limited database. In order to get a natural and well understandable articulation of the lyrics, the continuous or naturally abrupt timbral variations occurring on co-articulation parts between pairs of phonemes should not be altered. For this reason, the basic segmental unit used in our concatenative system is the diphone, which consists in a sound sample of 2 consecutive phonemes, as was already used in early concatenative speech synthesizers [MC90; Dut+96] and in more recent singing synthesizers [Bon08a; KO07]. In order to synthesize any possible lyrics, the minimum requirements for the database is to cover all the possible diphones of the language to be synthesized (also referred to as "di-allophones" in [Bon08a]). In this thesis, we focused on the French language, which requires about 1200 diphones (for 36 different phonemes considered, listed in annexe A). In some systems, longer units like triphones may also be used to increase the quality for some articulations [Ken12], but this is not a requirement. For sustained note, it is also useful to integrate in the database long sustained vowels that may be used to limit the requirement for time-stretching.
This complete phonetic coverage has to be done for at least one pitch and one intensity value. In order to minimize the transformations during synthesis, the chosen pitch should be chosen inside the usual range of the singer. Due to the quality of transposition that is usually better for upward transpositions (as discussed in sections 2.4.2.1 and 3.5.1.2), it may be preferable to choose a pitch lying in the lower part of the singer's pitch range. The intensity of the database should be medium (neither too soft nor loud), at a level that is comfortable for the singer, and the timbre should be "neutral" (without any specific expression or unusual vocal quality). The speed has some influence on the articulation of phonemes, and could be also considered as an additional parameter to be taken into account in the database, as has been done in [Bon08a]. For a single speed coverage, it is preferable to choose a rather regular and slow speed, so that the need for time-stretching transformations is minimized (stretching short units to create long ones is harder than the inverse process).
Then, once those minimal requirements are satisfied, the database can be extended by recording it several times with various combinations of pitch, intensity, speed and timbre. Although the memory availability is less and less of a problem nowadays, covering several of those combinations can quickly increase the quantity of data to be managed and the work load required to record, format, annotate, and analyze it properly. Although the annotation and analysis work can be partly automatized, some non-negligible manual work to correct the annotations remains necessary for obtaining a good quality synthesis. Some third-party companies are specialized in creating and selling databases to be used with softwares like Vocaloïd [1] for amatory or professional musical production, in which case they can

[1] https://www.vocaloid.com/en/products

spend much effort in the creation and annotation of such databases. However, some users may want to record their own database with a specific voice, in which case it is advantageous to minimize its size. For this purpose, a possible compromise is to record only sustained vowels on several pitch and intensities (and timbre), in order to extend a bit the covered space without requiring much more recordings, as vowels represent the main part of the voice in singing.

3.2.1.2 Recording constraints

In addition to the specifications of the database, some technical constraints also have to be considered, relating to the recording process.
A first aspect is that when singing for a long time, a singer may get tired, and this may have some influence on his voice timbre. He/she also may have more difficulties to maintain a stable pitch all along. The longer the database, the longer the recording session, and the less stable the timbre and f_0 might be. A possibility to overcome this might be to record the database in several sessions, but the room, microphones positions and gains used for recording should not be changed. Splitting the session would also increase the risks for the singer to have a different timbre (e.g: if his/her voice is not well heated, if he/she is more tired, if he/she got cold and has a husky voice, ...).
For synthesis, we need an homogeneous database in order to minimize the timbre differences between the concatenated units. For these reasons, the length of the database should thus be minimized so that it can be recorded in a single session without being too intensive for the singer.

3.2.1.3 Recording script

For covering all the necessary diphones in the database, a script has to be established, as diphones segments can't be sung alone, isolated, without a minimum of context. A simple solution for this is to use a systematic approach. For instance, the singer may sing all combinations of type $_CVCVC_$ (e.g. "babab"), $_V_1V_2V_1_$ (e.g. "aoa") and $_VC_1C_2V_$ (e.g. "abda"), where "_" is a silence, "C" designates a consonant, and "V" a vowel. This strategy was the one used for a first prototype of our system developed during an internship just before this thesis, as explained in [Ard13]. A rather similar approach was used in [Mac+97a], minimizing the size of the database using C_1VC_2 tokens. But this is not very natural to sing, and this systematic approach is also not very optimized in term of length, as it has quite a lot of redundancy.
We thus chose for our databases to use real words. Our project partner Acapela Group[2] established a recording script matching those constraints by using a greedy optimisation algorithm that tries to cover all diphones using a minimum number of words from a dictionary. Some constraints were imposed to the algorithm in order to avoid choosing words that are too long or too complicated to pronounce. This resulted into a list of around 900 French words. Some rare diphones may not be found in isolated French words, but might still exist when chaining 2 words together. A few pairs of words were thus also selected instead of single words to cover those cases.

[2] http://www.acapela-group.com/

Word	Phonemes	Diphones
coïnculpé	_ k O e~ k y l p e _	[_k, kO, Oe~, e~k, ky, yl, lp, pe, e_]
ovni	O v n i	[_O, Ov, vn, ni, i_]
myosotis	_ m j O z O t i s _	[_m, mj, jO, Oz, zO, Ot, ti, is, s_]
parking lapin	_ p a R k i N l a p e~_	[_p, pa, aR, Rk, ki, iN, Nl, la, ap, pe~, e~_]

TABLE 3.1: Example of words from the database's textual script along with their phonetic transcription and corresponding diphones (in SAMPA notation)
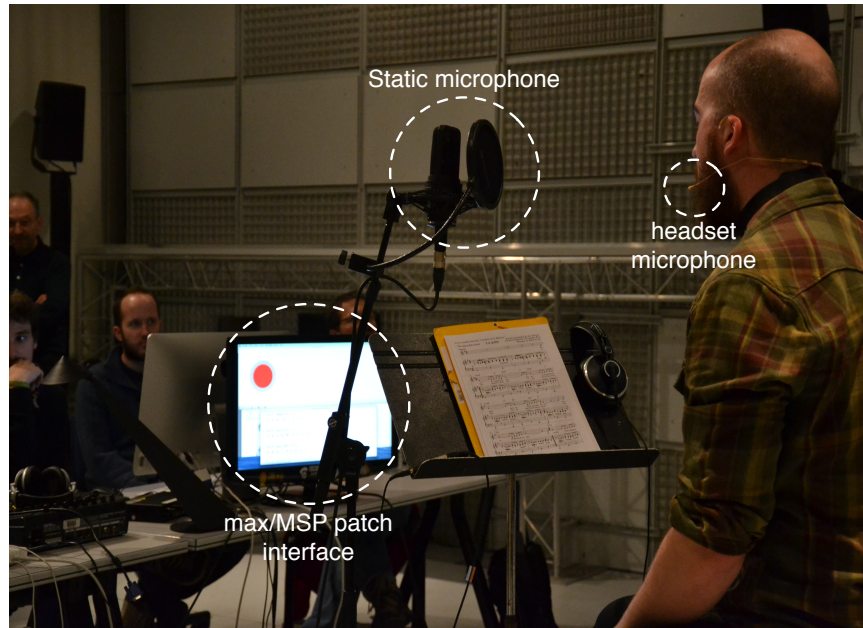
FIGURE 3.2: Recording set-up

Table 3.1 gives a few examples of words from our database along with their phonetic transcription and the diphones covered by these words.

3.2.1.4 Recording process

The databases have been recorded in a studio at IRCAM, using 2 microphones. The 1st one was a headset microphone (DPA4066), and the 2nd one a static microphone (AT4050) placed at a distance of about 1m in front of the singer. Figure 3.2 shows a picture from a recording session with this microphone set-up.

The instructions given to the singer were to sing each word of our script at a stable pitch and intensity level, at a constant and reasonably slow rate (around 60BPM with one syllable per beat). The singer was also instructed to try avoiding vibrato as much as possible, as vibrato induces small timbre and intensity variations (tremolo) that would remain when transposing the pitch and are thus not desirable. However, it is less natural and thus more difficult and tiring for the singer to sing without vibrato. Some amount of vibrato is thus still present in the recordings, but it remains limited.

In order to help with the recording process, a patch (program) has been developed in max/MSP[3]. Its aims at monitoring the correctness of the pronunciation of

[3] https://cycling74.com/products/max
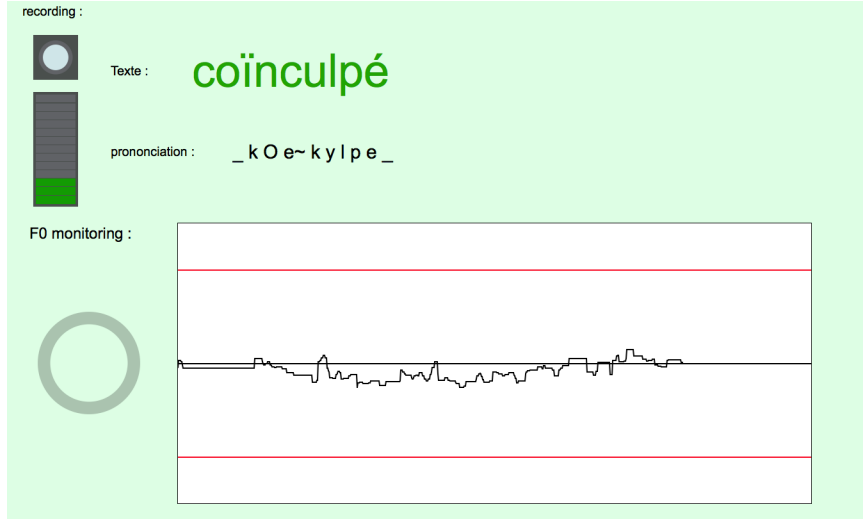
FIGURE 3.3: Singer's view of a max/MSP patch used for monitoring recording sessions.

TABLE 3.2: Summary of the recorded databases

Database	Sex	Range	Style	Database pitch	speed	vowels	
						pitches	intensities
RT	male	tenor	pop / variety	D3 (145H_z)	~60BPM	C2	pp, mp, mf, f, ff
MS	female	mezzo-soprano	pop / variety	D#4 (315H_z)	~60BPM	B3, F4, D5	pp, mf, ff
EL	female	soprano	lyrical	A4 (440H_z)	~60BPM	C4, B4, F5	pp, mf, ff

the singer, while helping him/her to keep a stable pitch and intensity during the whole session. Figure 3.3 shows the singer's view of this patch. It displays the current word to be sung along with its phonetic transcription, so that the correctness of the pronunciation can be verified. The singer's f_0 is also displayed in real-time along with the target pitch value (black horizontal line), using a real-time implementation of the yin algorithm [CK02]. The red lines represent a margin around the target f_0 inside which the singer's f_0 should remain. Similarly, a level meter helps to monitor the intensity level throughout the whole session. For each word, the current time of the recording is also saved to help with the automatic segmentation of the database.

3.2.2 Description of recorded databases

We recorded 3 databases in the course of this thesis. First, we chose 2 professional singers: 1 tenor male pop singer, and 1 female mezzo-soprano pop singer. In addition, we recorded a 3rd database from a female soprano lyrical singer to be used for a piece from the composer Arnaud Petit, using our system. In the following, those three databases will be labeled respectively RT, MS and EL. Additionally, sustained vowels on several pitch and intensity combinations have been recorded for each database. Table 3.2 summarizes the characteristics of those 3 databases. Some example sounds from the 3 databases are given in sounds 3.1 to 3.6.

The total length of the sound files (including some silent part around sung words) for one database is around 1h30, and each database could be recorded in half a day.
A system has been integrated to easily extend the database with "add-ons" that

can be used in addition to the main database by the unit selection system, in order to cover additional pitch and intensity values, or additional language-specific phonemes (e.g. to synthesize English lyrics).

3.2.3 Database annotation

In order to be used by a synthesis system, the databases must first be properly annotated to define the units to be concatenated and further adjust their durations. The annotation and unit-selection strategy may differ depending on the system. For instance, in [Mac+97a], variable-size units are used based on a simple phonemes segmentation. In our system, the basic units used are diphones, and several segmentation levels are used.
In order to delimit the diphones units and be able to adjust phonemes durations, a first segmentation step into phonemes is necessary. This segmentation was done automatically by our project partner Acapela Group[4], using an automatic speech recognition program to align the recorded sound with the phonetic transcription of the corresponding words from the script. This segmentation was then verified and manually corrected. From this phonemes segmentation, a first simple strategy to obtain the diphones is to cut the phonemes in the middle.
However, due to co-articulation, phonemes don't have constant timbre characteristics. In voiced phonemes, the formants move from one position to another between 2 phonemes. Plosive consonants /p/, /t/, and /k/ are made of 2 distinct parts: a silent part followed by an explosion that is characterized by a short burst of broadband noise. For each phoneme, one can thus distinct the co-articulated part where the timbre is changing at the beginning and end of the phoneme, and a stable part where the timbre is almost constant. This stable part is the one that is hold when the phoneme is sustained for some time, which happens much more in singing than in speech. In the case of a plosive that is particularly lengthened (e.g. for expressive purpose to give more emphasis on a syllable), the sustained part would be the silent part which thus constitutes the stable part in this case.
For unit concatenation, the phonemes should be connected in the stable part, where the timbres are similar for both units. In most cases, the middle of the phonemes belongs to the stable part, and our first strategy for delimiting diphones should thus give a satisfying result. However, in some cases (e.g. for some plosives), this might not be the case, and this automatic diphones segmentation may thus also have to be manually corrected to avoid artifacts related to an inappropriate segmentation.

In our system, a 3rd level of annotations for delimiting the stable parts defined above may be used optionally. Although not absolutely necessary, this additional segmentation has several advantages. For long sustained notes, some steady vowels recordings (or "vowels kernels") are inserted as additional units in-between the diphones in order to match the target phoneme's durations and minimize the need for time-stretching. But for shorter phonemes, units have to be shortened using time-streching (or rather "time-shrinking"). Using the stable parts annotations, it is possible to adjust the boundaries of the selected units to adjust their length according to the target phonemes' lengths and thus minimize the need for time-stretching, without affecting the co-articulated parts which are delimited by this annotation. Such markers where also used in [Bon08a] for similar reasons. This annotation can also allow to apply different time-stretching factors to the co-articulated and stable parts, e.g. for obtaining very sharp or smooth articulations

[4] http://www.acapela-group.com/

by changing the duration of the co-articulated parts without changing the total phoneme duration, which may be used as an expressive feature. This option has been integrated into *ISiS* and can be configured to apply a specific time-stretching factor for co-articulated parts. Finally, in case sustained vowels are available with several pitches and/or intensities, the stable parts annotation can also be used in order to select short diphones that encompass only the co-articulation, as was proposed in [Ard13], and sustained vowels close to the target pitch and intensity can be inserted in-between to occupy most of the vowel's duration. However, these segmentations require fastidious manual correction to be really usable, while the simpler strategy using only the diphones annotation can already give rather good results.

Based on those annotations, one can then select in the database the samples to be concatenated and define the time-stretching factors for matching the desired phonemes durations given by the control module. Note that only diphones and sustained vowels are used for now, but the phonemes and stable parts segmentation strategy may also allow the use of longer units that may be considered for later improvements. Figure 3.4 shows as an example a spectrogram of the word "ovni" from the database with the 3 annotation layers.
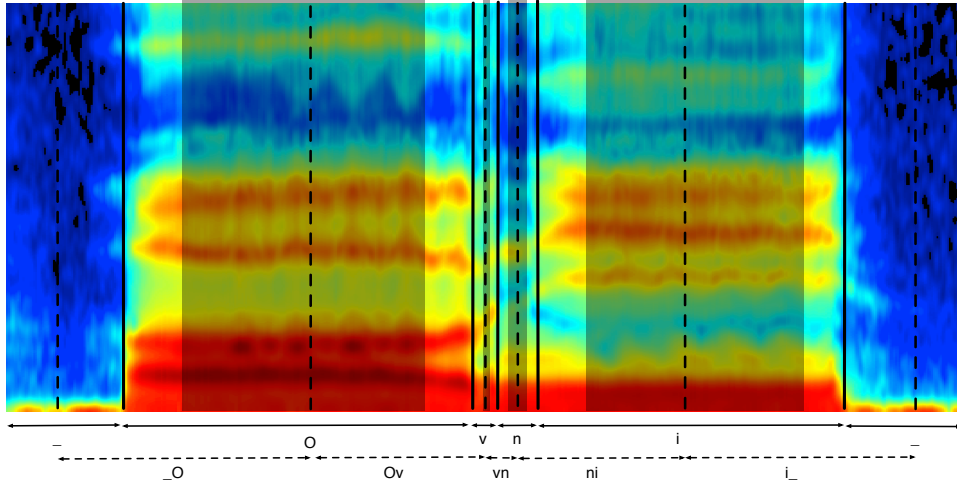
FIGURE 3.4: Example of database annotations for the word "ovni", showing the 3 annotation layers. Plain vertical lines delimit phonemes, dotted lines delimit diphones, and shaded areas show the stable parts. The spectrogram shows the True-Envelope analysis.

3.3 Units selection

Regarding the unit selection process, the input of the system, obtained from the control module, are a sequence of phonemes and their associated durations, as well as target pitch and intensity values. From the phonemes sequence, it is thus first necessary to group them into a sequence of units' labels. Diphones' labels are simply obtained by grouping couples of successive phonemes together (including silences). For each vowel, a sustain unit is also inserted in the middle, that will be used in case of long notes. Table 3.3 gives an example of a phonemes sequence

and the corresponding sequence of labelled units.

TABLE 3.3: Example of a phonemes sequence and corresponding units labels.

Raw text	"C'est une chanson qui nous ressemble."
Phonemes sequence	_ s, E, t, y, n, S, a~, s, o~, k, i, n, u, R, 2, s, a~, b, l, @, _
Units labels	_s, sE, E, Et, ty, y, yn, nS, Sa~, a~, a~s, so~, o~, o~k, ki, i, in, nu, u, uR, R2, 2, 2s, sa~, a~, a~b, bl, l@, @, @_

The role of the unit selection module is then to select in the database the best sequence of units to be concatenated (as explained in section 2.3.3), according to the given labels. As said previously, the script established for the database recordings aimed at minimizing the quantity of necessary recordings. As a result, many diphones only appear once in the database, which doesn't leave much options to the unit selection system. Obviously, the role of this unit-selection module is thus much more restricted than in speech synthesis systems where large databases with variable-size units are used. However, there still is a part of redundancy in the database, which requires some means to choose the best unit when there are several possible choices.

MFCCs [DM80] are a compact way to describe the timbre of a sound frame, that is typically used in unit selection [Vep04]. It basically consists in passing the DFT of the signal into a filter bank of triangular windows equally spaced along a Mel-warped frequency scale (a perceptually-motivated frequency scale to approximate the behaviour of the auditory system), and compute the cepstrum of the result. In order to minimize the timbral differences between consecutive units, we thus defined a concatenation cost as a simple euclidean distance between the MFCC coefficients of the left and right units, computed at diphones' boundaries, on the stable parts of the phonemes.

Optionally, in case several pitches have been recorded for the sustained vowels (in our case for the MS and EL databases), a target cost C_t is also defined as:

$$C_t = w_{d_{cents}} \cdot |d_{cents}| \cdot \frac{10}{1200} \quad \text{with} \quad \begin{cases} w_{d_{cents}} = 1 & \text{if} \quad d_{cents} < 0 \\ w_{d_{cents}} = 0.5 & \text{if} \quad d_{cents} >= 0 \end{cases} \tag{3.1}$$

where $d_{cents} = 1200 \cdot log2(\frac{f_{target}}{f_{orig}})$ is the pitch difference, in cents, between the mean pitch of the considered unit f_{orig} and the pitch f_{target} of the target note to which the phoneme belongs. The weight $w_{d_{cents}}$ is used to favour upward transpositions over downward transposition.

The Viterbi algorithm [For73] is then used to find the best sequence of units that gives the lowest overall cost. Better approaches to unit-selection probably exist, as proposed in [KV98; Vep04]. However in our case, due to the limited choice of possible units from the database and an important use of transformation techniques, the impact on the synthesis result would probably be rather limited, and this issue has thus not been much investigated.

Then, once the units have been chosen, the segments boundaries have to be adjusted according to the phonemes' durations, as will be explained in the following section.

3.4 Time corrections

As explained above, our system is based on the concatenation of diphones and stable vowels. The diphones units are composed of 2 half-phonemes, and the phonetic annotation from the database gives us the boundary between those 2 half-phonemes in each diphone unit. After concatenation, a full phoneme is thus composed of up to 3 parts: the right half-phoneme of the left unit, the left half-phoneme of the right unit, and possibly a sustain unit in the middle for vowels. In this section, we will denote the durations of these 3 parts respectively by d_1, d_2, and d_3, as shown in figure 3.5. For each phoneme, the total duration after concatenation d_{concat} is thus obtained by summing the durations of these 3 parts ($d_{concat} = d_1 + d_2 + d_3$).
Two means are available to ensure that the final phonemes durations in the synthesis match the target phonemes durations given by the control module: adjusting the boundaries of the selected units, and using time-stretching to further extend or reduce their durations. Those 2 means are used in our system in complement of each other, as will be explained in this section.
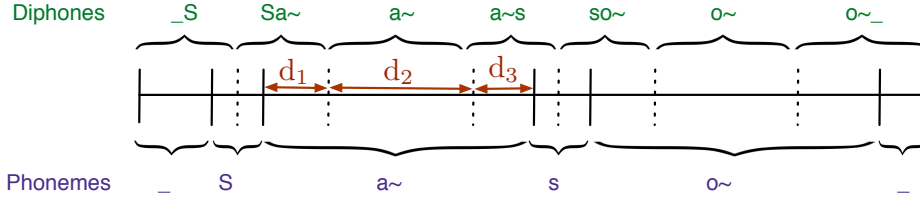
FIGURE 3.5: Relation between phonemes and concatenated units for the synthesis of the French word "chanson", illustrating the segments d_1, d_2, and d_3 on the phoneme /a~/

In order to avoid the possible degradations that may arise due to the use of time-stretching transformation, the units' boundaries are first adjusted at best according to the target duration of each phoneme d_{target} while minimizing the need for time-stretching after concatenation. Slightly different strategies are used, depending if the stable parts segmentation is used or not.

3.4.1 Without the use of stable parts

In case the stable parts are not used, the following rules are applied to adjust the segments boundaries for vowels:

- If $d_{concat} < d_{target}$, which may happen in case of long sustained notes or melisma (group of notes sung on a single syllable), then the units boundaries are not adjusted, and only the middle segment of the sustained vowel is stretched so that the co-articulation parts are not altered. The stretching ratio for this segment is thus $r_{d2} = \frac{(d_{target} - (d_1 + d_3))}{d_2}$.

- If $d_{concat} > d_{target}$ and $d_{target} > d_1 + d_3 + d_{min}$, then the middle segment is shortened, thus adjusting d_2 so that $d_{concat} = d_1 + d_2 + d_3 = d_{target}$. The value d_{min} is the minimum duration for a selected unit to be used in the concatenation, used here to avoid using a too short segment for the middle unit.

- If $d_{target} <= d_1 + d_3 + d_{min}$, then the middle segment is dropped ($d_2 = 0$) and d_1 and d_3 are adjusted such that $d_1 + d_3 = d_{target}$. For this purpose, the

ending time of the left diphone is decreased and the starting time of the right unit is increased by the value $\delta_d = \frac{(d_1+d_3-d_{target})}{2}$.

The times boundaries adjusted using those rules are then used to finally extract from the database the segments to be concatenated.
For consonants, there is no central stable unit (d_2), and the units boundaries are not adjusted, as without the help of a stable part annotation, the boundaries of the selected units can't be properly adjusted without risking to alter the pronunciation.

With the described strategy, the co-articulation parts of vowels are thus never stretched neither compressed. We assume that this corresponds to what happens in the reality, to some extent: when we speak fast, because of the co-articulation, vowels don't have enough time to reach their stable position and are constantly transiting between the preceding and following phonemes.
However, this strategy may cause problems for very short notes, as the segments may become too short for the phoneme to be recognizable and thus the lyrics to be understandable. For this reason, a minimum duration d_{concat}^{min} is used for the concatenation of vowels such that we always make sure that $d_1 + d_3 >= d_{concat}^{min}$, where d_{concat}^{min} has been empirically set by default to 0.2s (for vowels). Then, if $d_{target} < d_{concat}^{min}$, a compression ratio $r_{comp} = \frac{d_{target}}{d_1+d_3}$ is used to match the target duration by compressing the 2 segments. This compression ratio is also used in the case of consonants with short target durations.

For expressive purposes, it may nevertheless be desirable to have a sharper or smoother articulation that the one recorded in the database. An additional option thus allows to specify a fixed time-stretching factor r_{art} to be applied to the first and last segments d_1 and d_3 that contain the co-articulated parts. In case $r_{art} \neq 1$, the same rules as described above are used, where the values d_1 and d_3 are replaced by $d_1 \cdot r_{art}$ and $d_3 \cdot r_{art}$.

3.4.2 With the use of stable parts

In case the stable parts annotation is used, the strategy for adjusting units is similar to the previous case, except that the segments d_1 and d_3 can also be extended in the limits of those stable parts. This reduces the need for the middle segment d_2 in some cases, which also reduces the number of concatenation points and thus the risks of timbral mismatch between concatenated units.
A last possible strategy is to set by default d_1 and d_3 to their minimal duration, given by the stable parts annotation, so that they only contain the co-articulated parts, and thus maximize the use of the segment d_2 to have a stable timbre and avoid concatenating in the central part of the sustained vowels.
In our system, those different strategies can be configured by the user. But it is difficult to assert which of those strategies is the best, as each has its own advantages and drawbacks, that depend on several factors like the availability and quality of a stable parts annotation, or the homogeneity and content of the database. The last strategy using the minimal durations for the segments d_1 and d_3 is especially interesting in case the database contains vowels recordings at several pitches, as the best unit can be chosen for the segment d_2, depending on the target pitch, and thus occupy the main part of the sustained vowel.

For consonants, the durations of the 2 segments d_1 and d_3 can be first adjusted inside the limits of their annotated stable parts to match the target duration, and time-stretching is thus used only if necessary (in case the target length is particularly short or long).

Note that for particularly long phonemes, an alternative possible approach to avoid using time-stretching would be to loop the stable part. But this possibility has not been implemented, as time-stretching ratios are usually small enough to avoid audible artifacts.

3.5 Synthesis engines

Once the units have been selected and their time boundaries properly adjusted, a synthesis engines is in charge of generating the synthesis. The synthesis engine has several purposes: it is in charge of concatenating the segments while smoothing the discontinuities at junctions in order to obtain a continuous signal comparable to a real voice, with a minimum of artifacts, as well as transforming the units in pitch and duration according to the input control parameters.
As explained in section 2.4, different approaches may be used for signal modeling. Depending on the approach used, some expressive timbre variations (intensity, roughness, ...) may also be already applied during the synthesis, or alternatively the synthesized sound output by the synthesis engine could be transformed in a second step. However, the problem of applying such expressive transformations will be addressed in chapter 6, and this section will only focus on the problem of generating an intelligible voice that follows the melody, rhythm and lyrics given as inputs. In our *ISiS* system, 2 independent synthesis engines have been integrated and can be used for synthesis with the same database and control inputs. The SVP engine is based on a phase vocoder using the superVP software[5], and the PaN engine (for **P**ulse **a**nd **N**oise) is based on a parametric representation of voice using the LF source model, similarly to the SVLN and PSY methods described in section 2.4.3.3. We will review in the following sections the specificities of those 2 engines.

3.5.1 SVP engine

The phase vocoder, as explained in section 2.4.2.1 relies on an STFT analysis of the signal, that is used to transform the signal in pitch and duration while ensuring the coherence of the partials' phases from one frame to another when applying some transformations. In our system, we used the phase vocoder implementation of superVP [LR13] with the SHIP algorithm [Röb10] for high-quality transformations. The advantage of using the superVP software is that it already integrates all the necessary algorithms for signal analysis and transformations, ready to use for our purpose. (Note that a basic prototypic implementation of this synthesis engine had been already developed during an internship before the start of this thesis [Ard13], which has since then been further improved.) Apart from having an efficient and refined phase vocoder implementation readily available in superVP, an advantage of this approach is that it is more flexible than time-domain approaches for applying advanced transformations, while allowing a perfect reconstruction of the signal from the STFT when no transformation is applied (which is not the case for sinusoidal models or other parametric approaches). In this section, we will first

[5] http://anasynth.ircam.fr/home/english/software/supervp

shortly review the approaches used for pitch-shifting and time-stretching, along with the necessary analysis to be run on the database. Then we will explain in more details the specific processing steps related to the concatenation process and the use of the SHIP algorithm.

3.5.1.1 Time-stretching

In the SVP engine, the classical approach to time-stretching described in section 2.4.2.1 can be used. As previously explained, this approach basically consists in moving the positions of STFT frames and overlap-adding them. But for a better quality, we used the SHIP algorithm [Röb10] introduced in section 2.4.2.1, that reduces the phasiness effects by preserving the initial waveshape of the original signal (thus avoiding partials phases desynchronisation). More details on this algorithm will be given in the section 3.5.1.5.
Additionally, superVP integrates an algorithm to enable transients preservation in order to better preserve the quality of some consonants when stretching is applied [Roe03], especially useful for plosives to avoid transients smearing.

3.5.1.2 Pitch-shifting

As explained in section 2.4.2.1, 2 different approaches can be used for pitch-shifting using the phase vocoder: either using a first time-stretching step followed by some time-domain resampling; or directly manipulating each frame in the frequency domain to shift the sinusoids to new frequencies. Both approaches are implemented in superVP and give rather close sound qualities. However, while the first approach using time-domain resampling is appropriate for processing continuous sounds, it is not in the case for concatenated segments. Indeed, due to concatenation, the frequency of successive frames should already match before the overlap-add step to avoid discontinuities, which is not ensured by this technique as the transposition is obtained by resampling the signal after applying the overlap-add operation, and the overlapping frames at segments' junctions may thus have a different f_0. We thus use the 2nd frequency-domain approach in our work, that transforms each frame independently, so that the f_0 of successive frames is already coherent before applying the overlap-add process. Note that another advantage of this frequency-domain approach is that the computation cost is constant whatever the transposition factor, whereas for the resampling approach, this cost linearly increases with the transposition factor.
In addition, spectral-envelope preservation is used in order to keep the timbre as close as possible to that of the original sound. In our system, this is done by inverse filtering the signal with the estimated spectral envelope before applying the transposition, and applying it back afterwards.
The transposition factor is computed, according to the target pitch, based on a pre-computed f_0 analysis of the database. In unvoiced parts (e.g. unvoiced vowels like fricatives or plosives), the target f_0 value is linearly interpolated to have a continuous curve, and the transposition is thus applied on the whole signal. To some extent, we assume that this corresponds to a natural behaviour of the voice, as when the pitch is raised, the spectral centroid on noise segments such as fricatives also tends to increase. This also avoids creating artifacts (jump of the resulting f_0) due to possible voicing estimation errors at the boundaries of voiced parts. The original f_0 value used for computing the transposition ratio on unvoiced parts is the

closest valid f_0 value in the considered unit.
Like for the time-stretching, the SHIP algorithm is also used for pitch-shifting.

We already mentioned in paragraph 2.4.2.1 the possible artifacts related to transposition with the phase vocoder (especially present for downward transpositions), mainly related to a possible shift of noise regions into formants increasing the hoarseness, a lack of high-frequency partials, and to limitations of the spectral envelope estimation (although not specific to this approach).

3.5.1.3 Signal analyses

3 types of analyses are necessary for the SVP engine:

- STFT: The STFT analysis may be computed offline and stored in files to save computation time at synthesis. But because of the overlap between frames and oversampling factor used to get a good resolution, this requires a lot of memory space. It is is thus preferred to compute the STFT only for the segments used during synthesis by storing it in temporary files. For this STFT analysis, a Blackman window is used (which results in low side-lobs). The window size is equal to approximately 4 periods (based on the mean $f_0$value), rounded to the next power of 2 (in samples), with a minimum 0.015s, and the step size is set to a quarter of the window size.

- f_0: For estimating the f_0, we used a monophonic version of the algorithm described in [YRR10], implemented in superVP. This algorithm evaluates the plausability of several f_0 candidates with several criteria, using harmonic matching based on a sinusoids plus noise model. The details of this algorithm are however beyond the scope of the present work and thus won't be presented here. The window size is set to (at least) 4 periods, based on a given minimum f_0 value. For identifying the unvoiced part where the f_0 should be interpolated, a confidence score output by the algorithm is used to estimate voicing. By setting a threshold on this confidence score, all frames with a value under this threshold are considered as unvoiced. Although the algorithm used gives a reliable estimation in most case, the f_0 curve may sometimes be slightly manually corrected to avoid artifacts due to an estimation error (especially on some consonants with low energy and harmonicity, or at words' boundaries next to silences). The correction can be done for instance using the audiosculpt software [6].

- Spectral envelope: In our work, we use an implementation of the True-envelope algorithm available in superVP [RR05a]. As suggested in [RR05a], the optimal order $0.5\frac{F_s}{f_0}$ is used for the estimation, based on the estimated f_0 values, where F_s is the sampling frequency of the sound.

All the analysis were stored in separate files using the SDIF (**S**ound **D**escription **I**nterchange **F**ormat) format [7].

3.5.1.4 Units concatenation

For the concatenation, the STFT analysis of each units are first copied one after another into a new SDIF file. SuperVP then takes this concatenated analysis file as

[6] http://forumnet.ircam.fr/fr/produit/audiosculpt/
[7] http://sdif.sourceforge.net/

input to stretch and transpose the sound. The sound is resynthesized by superVP using an IFFT and overlap-add of this transformed STFT. The f_0 and spectral envelope analysis used for the transformations are also concatenated in separate files.
But, as the concatenated segments are usually not contiguous in the database, they are likely to present different characteristics that may cause audible discontinuities at segments' junctions, that would degrade the resulting synthesis. These discontinuities may be related to 2 factors: the spectral envelope and the phases. This problem is evoked for instance in [BL03] (which uses the EpR model for synthesis), and the proposed solution to smooth these discontinuities is to spread out the phase and spectral envelope discontinuities on a set of frames around the junctions.
We also propose here some solutions, specific to our system, that we implemented to smooth out such discontinuities. Note that these solutions were already implemented in the 1st version of the SVP engine, before the start of this thesis, as presented in [Ard13; ADR15]. For the sake of completeness, we will nevertheless give here some details on those solutions, implemented in our system.

3.5.1.5 Shape-invariant processing and phase correction

For minimizing the possible phasiness effects due to the vertical de-synchronisations of the harmonics' phases (related to the pulse's shape) when applying signal transformations on voiced sounds with the phase vocoder, we used the SHIP algorithm [Röb10] implemented in superVP, already introduced in section 2.4.2.1.
As previously explained, the SHIP algorithm can be seen as a frequency-domain SOLA ([RW85]) algorithm, where the phase of all partials are adjusted simultaneously with a similar time delay to maximize the cross-correlation between successive frames while avoiding a vertical phases de-synchronisation. This phase shift simulates the displacement of the window that is applied in the regular time-domain SOLA algorithm, but without actually displacing it.
As the same shift is applied to all partials "as a block", only the phase of the harmonic having the strongest amplitude (and thus the most impact on the computation of the cross-correlation) might be really well adjusted. But for a continuous sound, one may assume that the glottal source's shape will evolve smoothly and that the vertical phase alignment should be approximately constant from one frame to another. For this reason, all the harmonics' phases should thus be rather coherent after applying the phase shift.
But in the case of concatenated segments, nothing ensures that the vertical phase alignment is the same between the frames at the boundary of each segment. We thus can't just use the SHIP algorithm as is, as important phase discontinuities may thus arise for some harmonics, which would result into annoying audible artifacts. This problem is illustrated in figure 3.6, which shows the temporal evolution (waveshape) of 5 harmonics for 2 concatenated segments with the same amplitudes but different vertical phases alignments.

The figure represents schematically how the 2 segments would overlap, after the phase correction of SHIP has been applied. As one can see, the phase of the fundamental, which has the biggest amplitude is well adjusted and the sinusoid overlap coherently with the previous segment. By chance, the phases are also rather coherent for the harmonics 1 and 4. But for the harmonics 2 and 3, the sinusoids on the left and right segments are out of phase, which will create destructive interferences on the overlapping part. While the rest of the signal, outside of the overlapping part, is not altered, these phase interferences may create audible artifacts at the junctions between the segments if the phase gaps are too important. An example of this effect
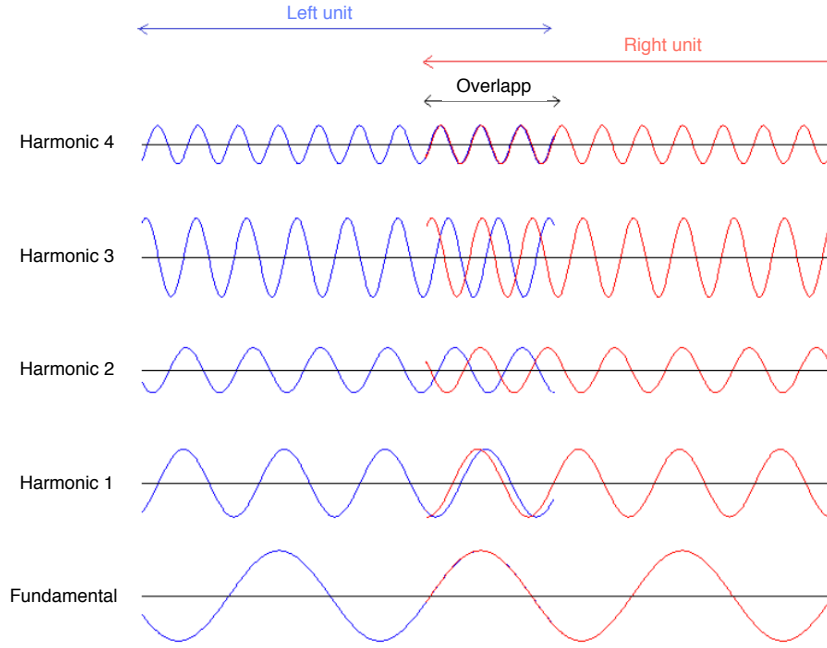
FIGURE 3.6: Schematic of 2 overlapping segments with different vertical phase alignments, when applying the SHIP algorithm

can be observed on the spectrogram on the left part of figure 3.7.
If the more classical phase vocoder approach is used instead of the SHIP algorithm, the phases of each harmonic are adapted independently of each other, which avoids this problem of interferences. But the shape invariance property of the SHIP algorithm would then be lost, and one might hear some phasiness. The solution we proposed is thus to use the SHIP algorithm, but spreading the phases differences along time (on an undetermined number of frames) when they are too important, so that the resulting phase gaps are small enough to be imperceptible, thus switching smoothly from the vertical phase alignment of the left segment to that of the right one.

It is thus first necessary, for each frame, to compute the vertical phase differences between the harmonics. Since, for each sinusoid, the phase evolution is proportional to its frequency, those phase differences are constantly evolving. But since the voice produces harmonic sounds, those phase differences repeat periodically. In order to be able to compare these relations of the phases from one frame to another, it is thus necessary to determine an anchor point where they should be computed. We chose to use for this the position where the phase of the fundamental is equal to 0. We use for this purpose the sinusoidal model obtained from the analysis done by superVP. Theoretically, voiced sounds being harmonic, only the harmonics should thus be represented by the sinusoidal model (apart from rough sounds that may also contain sub-harmonics, but that are not considered here). But it may happen that some frequency bins between the harmonics might be misclassified as sinusoids. Based on the estimated f_0 value, we thus only retain the sinusoids that are harmonically related to the fundamental (with a small margin for safety). For each harmonic, the phase at the center of the frame, obtained from the STFT analysis, is comprised in the $[-\pi;+\pi]$ interval. One thus need to compute the time lag d_t between the center of the frame (t_{center}) and the time where the fundamental

has a phase equal to 0 (t_0). This is obtained from the following equation:

$$d_t = \frac{-\varphi_{center}}{2\pi \cdot f_0} \tag{3.2}$$

where f_0 is the fundamental frequency (in H_z), and φ_{center} is the phase of the fundamental at the center of the frame.
Once we know d_t, is is possible to compute the phase of each harmonic at the position t_0, using the formula:

$$\varphi_0^i = \mathrm{Arg}(\varphi_{center}^i + i \cdot 2\pi \cdot f_0 \cdot d_t) \tag{3.3}$$

where φ_0^i is the phase of the i^{th} harmonic at t_0 and φ_{center}^i is the phase of the i^{th} harmonic at the center of the frame. The Arg function gives the principal argument of the phase, wrapped in the range $[-\pi; +\pi]$. This is computed as:

$$\mathrm{Arg}(\varphi) = (\varphi + \pi)\%(-2\pi) + \pi \tag{3.4}$$

Once the value φ_0^i is known for each harmonic, one can compute the differences in the vertical phases alignments between successive frames with:

$$\Delta_{\varphi_0^i}(n) = \mathrm{Arg}(\varphi_0^i(n) - \varphi_0^i(n-1)) \tag{3.5}$$

where n is the index of the frame. If this value is superior to a given threshold for a given harmonic i, it is thus necessary to correct its phase in the frame n in order to reduce this difference. The following condition is thus applied:

$$\begin{gathered} \text{if } \left|\Delta_{\varphi_0^i}(n)\right| > \Delta_{\varphi_{max}} \\ \Rightarrow \varphi_0^i(n) = \mathrm{Arg}(\varphi_0^i(n-1) + \Delta_{\varphi_{max}} \cdot \mathrm{sign}(\Delta_{\varphi_0^i}(n)) \end{gathered} \tag{3.6}$$

where $\Delta_{\varphi_{max}}$ is the threshold (set by default to 0.1). This process is applied to each frame successively.
As the resynthesis is done based on the STFT analysis using an IFFT for each frame followed by an overlap-add operation, the phase correction must be done on the DFT bins themselves (and not only on the sinusoidal model). For this purpose, the same phase correction as given in equation 3.6 is applied to all the bins belonging to a same sinusoid. These bins are defined as those comprised in the interval delimited by the 2 minimums around the spectral peak related to the considered sinusoid. Figure 3.7 shows the effect of this phase correction on the synthesized signal, by comparing a portion of the spectrogram exhibiting such phase mismatch between 2 concatenated units, with (on the right) and without (on the left) applying this correction. As one can see on the right part of the figure, the interferences at the junction between the 2 segments have disappeared.

Unlike the solution presented in [BL03], this phase correction is independent from the pitch-shifting factor applied, and is only applied if the discontinuity is big enough ($> \Delta_{\phi_{max}}$).

3.5.1.6 Spectral envelope interpolation

The second possible cause of discontinuities is the spectral envelope. At junctions between 2 units, spectral envelopes are never equal, even though the concatenated
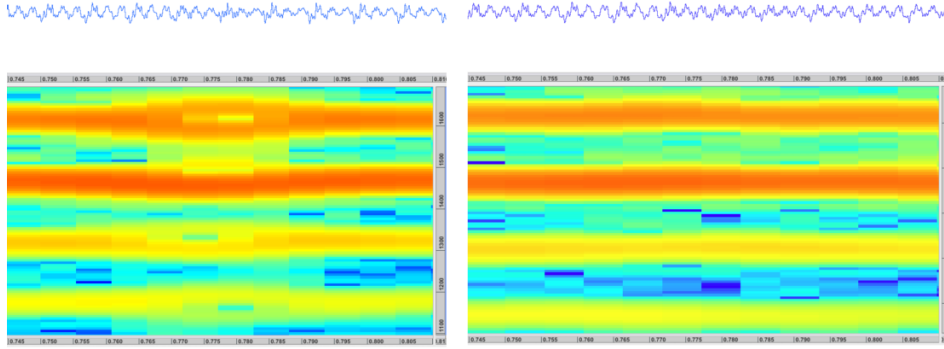
FIGURE 3.7: Spectrogram of 2 concatenated segments after resynthesis: without phase correction (on the left) and with the phase correction applied (on the right). The phase correction smooth out the phase discontinuities to avoid destructive interferences on overlapping sinusoids.

phonemes are the same and the unit selection system tries to minimize the timbral differences, because of the inevitable variability intrinsic to all natural sounds. For this reason, there may exist some disturbing timbral discontinuities at junctions that should thus be avoided. An obvious solution to overcome this is to interpolate the spectral envelope on a few frames around the junctions. For this purpose, we use a linear interpolation of the log amplitude of the estimated true envelope, which is sufficient as the envelopes are already rather close, such that more complex interpolation schemes don't make much audible differences. For vowels, this interpolation can be done on the whole stable part, as is suggested in [KO07]. However, this creates a very smooth envelope that lacks the small variations present in natural sounds, which may sound a bit more synthetic. We thus limit this interpolation on a limited duration around junctions (by default $min(0.1, 0.5 * d_{concat})$, in seconds). On plosives, no interpolation is done, as the junction point is contained in the silent part of the phoneme.
Once the spectral envelope has been interpolated, it is applied on the concatenated sound. Note that on stable parts, if the envelope is not interpolated, some timbral and amplitude modulations induced by the original vibrato of the recorded sample may be present, which may not be coherent with the new vibrato imposed in the synthesis, after transposition. Although the perceptual importance of these modulations is secondary compared to the frequency modulation of the vibrato, this may be slightly unnatural, which is why it is preferable to avoid having vibrato in the database recordings (although this is not always easy for singers).
Examples of copy synthesis (based on a real recording by RT) using the SVP engine are provided in sounds 3.7, 3.8, and 3.9.

3.5.2 PaN engine

Although the phase vocoder can provide good quality transformations, this approach also has some limitations. As explained previously, several artifacts may arise when important pitch-shifting factors are applied, which degrades the sound quality. Another limit is that the phase vocoder does not provide a mean to properly manipulate the glottal source parameters, which can be useful for instance to modify the intensity or breathiness of the voice, as evoked in section 2.5. It may be possible to use a filter to indirectly modify the spectral tilt or glottal formant (e.g. as was done in [ADC98] or [AD03]), but this does not allow to recreate

harmonics in the high-frequency when decreasing spectral tilt, neither to process the deterministic and stochastic components separately.
To overcome these issues, the use of a parametric approach where the glottal source parameters can be explicitly accessed is thus an interesting alternative. We presented in paragraph 2.4.3.3 the SVLN and PSY analysis/synthesis frameworks, which are such parametric methods. From the analysis of a voice signal, those methods allow to transform and resynthesize the voice, based on the source-filter paradigm.

Besides the SVP engine, a second synthesis engine, called PaN (for ***Pulse and Noise***), has thus been integrated in our synthesizer *ISiS*. This 2$^{\text{nd}}$ engine is based on the SVLN and PSY methods, with additional extensions to further improve the quality and flexibility of the transformations. Especially, it adds the possibility to change the duration and f_0 of the original signals, which was not yet possible with the initial implementation of PSY, as presented in [Hub15] for voice conversion purposes. Note that the implementation of the PaN engine, including the development of the extensions for pitch and time scale modifications, has been provided by Dr. A. Roebel who contributed to the research performed in the context of the ChaNTeR project. The related synthesis engine can therefore not be considered as a research contribution of the present thesis. While the PaN analysis/synthesis engine has not yet been published, its principle remains close to the SLVN and PSY methods that have been detailed in section 2.4.3.3.
For completeness, we nevertheless give here a short summary of the basic principles of this method, as this engine has been used for works on expressive voice transformations that will be presented in chapter 6. From the f_0 and R_d analysis, this approach basically generates, in the frequency domain, a stream of pulses with positions depending on the f_0, and shapes depending on the R_d values, based on the LF model. Using the R_d and VUF analysis, the VTF can be obtained from the spectral envelope estimated using the True Envelope algorithm, as explained in section 2.2.4. Each pulse can then be filtered by the VTF estimated at the corresponding position, transferred back to the time-domain using an IFFT, and finally overlap-added with the surrounding ones.

As the approach is parametric, transformations can be applied by simply manipulating the parameters from the analysis, rather than the signal itself. For time-stretching, the analysis curves (R_d, f_0, and VUF) can be time-scaled and resampled to match the target durations, according to the stretching factors determined as explained in section 3.4. For transposition, a simple factor is applied to the values of the f_0 curve. However, in our system, we can directly use the target f_0 curve obtained from the control module. The target pulse positions are deduced from the target f_0 curve, each pulse being spaced from the previous one by the fundamental period. The harmonicity value (confidence score) returned by the f_0 estimation algorithm [YRR10] is used to determine the voiced segments that should be synthesized using glottal pulses.
Regarding the noise component, the *ReMiDeMo* approach presented in [Hub15] is used to isolate it from the deterministic part. Then, it can be summed with the pulses to generate the final voice signal. For this purpose, a mapping is done between the original pulses' positions of the concatenated segments and the target positions, and the noise component at the original pulse position is windowed and copied in a pitch-synchronous manner from the database at the corresponding position in the synthesis. On unvoiced segments, artificial pulse positions are used

as target positions where to copy the windowed noise, with a period obtained by interpolating the f_0 values at the boundaries of the surrounding voiced segments.

3.5.2.1 Signal analysis

5 types of analysis are used for the PaN engine, and are therefore requested to be present for each file of the database:

- f_0: The f_0 analysis is similar to the analysis used for the SVP engine, using superVP. For each analyzed frame, a confidence score is returned by superVP, along with the estimated f_0 value. All values that have a confidence score below a given threshold are set to 0 in order to identify unvoiced segments.
- Spectral envelope: Similarly to the SVP engine, the True-envelope algorithm [RR05a] implemented in superVP is used to estimate the spectral envelope for the PaN engine.
- VUF: The VUF analysis is also implemented in superVP. It basically computes the ratio of the energy related to the sinusoidal peaks inside a given frequency band over the total energy of the frequency band to classify this band as voiced or unvoiced and the VUF is set to the highest frequency band having sinusoidal content [Hub15].
- R_d: The R_d source parameter is analyzed based on the approach presented in section 2.2.4, implemented in superVP.
- Noise component: The unvoiced component of the voice is separated from the sinusoidal part using the ReMiDeMo approach [Hub15] (introduced in section 2.4.3.3), and stored as separate sound files in the database.

Note that with this approach, as the pulses are artificially generated from the ground up, there is no phase issue related to concatenation. However, the spectral envelope still requires to be interpolated around junctions, similarly to the SVP engine. And this is also the case for the R_d analysis so that the pulse shape doesn't drastically change from one pulse to the next at junctions between segments. Similarly to the SVP engine, the same time boundaries determined as explained in section 3.5.1.6 are used for the interpolation of both the spectral envelope and the R_d parameter.
Examples of copy synthesis (based on a real recording by RT) using the PaN engine are provided in sounds 3.10, 3.11, and 3.12.

3.6 Summary

We presented in this chapter the system we developed to synthesize singing voices, offline from a score and lyrics. This system is based on concatenative synthesis and integrates 2 synthesis engines. We described the constitution of the databases used by the system, along with the various constraints and specification that need to be considered to produce a good synthesis. The 2 approaches used by our synthesis engines are the phase vocoder, and a novel parametric approach, based on previous frameworks for voice synthesis and voice conversion. The 2 approaches have different limitations in terms of signal manipulation and potential artifacts. In the framework of the ChaNTeR project, an evaluation has been run, comparing the 2 engines SVP and PaN of our system, along with other systems

developed by other collaborators of the project, which suggests that our 2 engines can generate synthesis with a similar quality, in terms of naturalness. The details of this evaluation are given in [Feu+16].
The PaN approach has interesting potential for expressive voice transformations, as it provides a direct access to the glottal source parameters that may be modified using specific rules for intensity and breathiness transformations, and allows a precise control of pulses positions and amplitudes that may be useful to introduce jitter and shimmer for generating rough voices. These possibilities will be further discussed in chapter 6.

Concerning this chapter, the contributions of the author in the presented work are:

- In relation with the other collaborators of the ChaNTeR projet, the choice of a strategy for the databases recordings
- The development of the max/MSP patch used in the recording sessions
- The establishment of a strategy for the segmentation of the databases. (The annotations themselves have mainly been done by collaborators from the Acapela Group company[8].)
- The development of the global architecture of the ISiS synthesis system
- The development of the units' selection module (partly undertaken during an internship, before the beginning of this thesis).
- The development of the SVP synthesis engine, and in particular the phases corrections and envelopes interpolations used to smooth out discontinuities during concatenation (partly undertaken during an internship, before the beginning of this thesis).

The databases and the SVP synthesis engine with phase and spectral envelopes interpolations have been succinctly described in a publication [ADR15]. Although the author of this work helped in the integration of the PaN engine into the ISiS system, the synthesis engine itself has been developed by Dr. Axel Roebel and thus can't be accounted as a contribution from this thesis work.

[8] http://www.acapela-group.com/

Chapter 4

Control module: modelization of the synthesis parameters

4.1 Introduction

In the previous chapter, we addressed the problem of generating an intelligible voice signal, based on units concatenation. In that part, we assumed that the inputs (f_0 curve and phonemes durations) of the system (unit selection + synthesis engine) where already precisely known. In the present chapter, we address the problem of generating these low-level features from the high-level informations of the score and the lyrics, which is the role of the control module, as illustrated on figures 1.1 and 2.11. We only focus here on the main control features, which are the f_0, the phonemes' durations, and the intensity. In the previous chapter, the intensity was not considered for generating the synthesis, as this will be the subject of a dedicated section in chapter 6.
We reviewed in section 2.6 the main models and approaches that may be used for generating the control parameters. The means to generate the f_0 and intensity curves can be divided in mainly 3 categories: parametric models (including some rule-based approaches), statistical approaches (based on HMMs or neural networks), and units concatenation. The chosen approach should be able to reproduce with sufficient details the parameters to carry the various fluctuations related to the naturalness and expressivity of real singing voices.

Although the problems of expression control and style modeling are intrinsically related, we will first present in this chapter some generic parametric models that we use for generating the control parameters for synthesis, and we will present in a next chapter an approach for learning and choosing the parameters of those models from a database to model specific singing styles.

In the present chapter, we start by shortly explaining how the phonetic transcription of the lyrics can be specified. Then, we present the rules and models we implemented in our control module to generate the 3 main control parameters to be used by the synthesis engine, which are the phonemes' positions and durations, the f_0, and the intensity, trying to reproduce the various fluctuations conferring to the voice its naturalness and expressivity while providing intuitive controls to the composer for shaping the expression. For this purpose, the control module is first used to automatically generate the parameters from a score and lyrics. Then, the parameters can be manually edited to refine the result.

4.2 Phonetic transcription

As we saw in the previous chapter, the unit selection module takes as input a sequence of phonemes along with their respective durations. But for allowing people to use our synthesis system without requiring any specific knowledge of phonetics, it should ideally enable to enter the lyrics as raw text. In such case, a program must be used to provide a phonetic transcription of these lyrics in order to further generate the sequence of units labels to be used by the units selection module. We use for this purpose the phonetizer "Liaphon" [Bec01] (developed at the " **L**aboratoire d'**I**nformatique d'**A**vignon"), already used in other softwares at IRCAM for audio and text alignment (ircamAlign [Lan+08]) and for text-to-speech synthesis (ircamTTS [OVL12]). This program takes as input the lyrics written in French and outputs the phonetic transcription of this text, that we then convert to the SAMPA phonetic alphabet [1].

However, such phonetizer programs have been developed for speech, but some problems arise when applied to singing voice. Contrarily to speech where the main focus is on the meaning of the text pronounced, the main focus in singing is on the pronunciation, which is further constrained by the melody and rhythm. For this reason, the text in singing must contain a specific number of syllables that is coherent with the number of notes in the score. Although a vowel can be sustained across several notes in case of melisma, the text can't contain more vowels than the number of notes in the score. In French, syllables may sometimes be eluded when speaking, whereas they should absolutely be pronounced in singing as it should carry a note (and the inverse may also be true). The output of the phonetizer may thus need to be manually corrected.
The ideal solution to overcome this problem would be to directly associate each syllable to a specific note, using a dedicated Graphical User Interface (e.g. as it is the case in the Vocaloïd software), and the phonetizer should take into account this syllabic division.
But as such interface has not yet been developed for our system, we usually directly input the sequence of phonemes in SAMPA notation (convenient to use as it only uses ASCII characters).

4.3 Timing

The rhythmical informations from the score provides the durations of the notes, to which the phonemes sequence should be properly aligned. The control module should also provide a duration for each phoneme that corresponds to a natural elocution, while being constrained by the rhythm. These durations mainly depend on the phoneme identity itself, but also on the singer's own elocution habits, and on singing style or specific expressive intentions, as phonemes durations can be used as an expressive mean to emphasize some notes.

4.3.1 Temporal alignment of notes and phonemes

While the basic linguistic unit used for expressing the alignment between the text and the notes in a musical score is the syllable, it is widely admitted that the notes'

[1] https://en.wikipedia.org/wiki/Speech_Assessment_Methods_Phonetic_Alphabet_chart
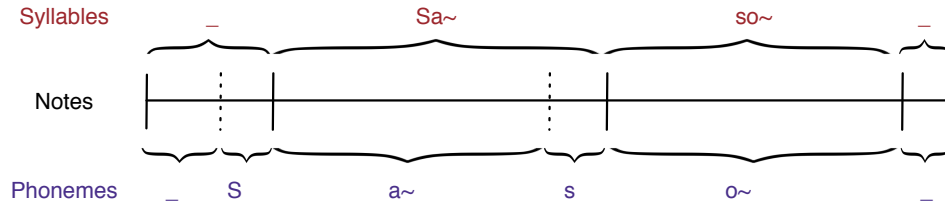
FIGURE 4.1: Illustration of the temporal alignment between the notes and phonemes for the French word "chanson", sung on 2 notes (with silences at the beginning and end)

onsets should actually rather be aligned with the vowels' onsets [Sun06; Mac+97b; Une02; Bon+01a; KO07; Bon08a]. We thus implemented this rule in our system as the main constraint for positioning the phonemes according to the given notes durations. The consonants that may be found at the beginning of a syllable should thus be contained in the duration of the note preceding that associated to the vowel of the syllable, as illustrated in figure 4.1, thus forming a "reversed syllable" with the preceding vowel.

Based on this simple rule, one thus start by associating each phoneme to a specific note of the score. A note thus contains at least a vowel, and the succeeding consonants if any (in case of a vowel sustained across several notes, the vowel should be repeated several times in the lyrics).

However, if this rule can be easily empirically verified, some cases seem more ambiguous regarding the temporal alignment. This is especially the case in French for syllables starting with a semi-vowel (/w/, /j/, and /H/ in SAMPA notation). Whereas the boundary between the consonant and the vowel is clear for consonants like plosives or fricatives, semi-vowels can be seen as a smooth transition between 2 vowels, and the boundary is thus less clear. In such case, the perceived onset of the note may be found somewhere during the course of the semi-vowel, before the actual vowel's onset. However, further investigations would be necessary to infer some rules for the notes-to-phonemes alignments in such specific cases, according to the phonetic context. Studies on the perception of rhythmical cues in speech, in relation to the articulation, have defined the perceptual center (or P-center) as the perceived attack point of each utterance [Fow79; Mar81], which may give useful indications to improve our alignment rules for singing. Meanwhile, we stick with our basic rule for aligning vowels to notes.

In case of a polyphonic song, it may also be necessary to introduce small random time lags for each voice, to avoid a perfectly similar alignment of the different voices for more naturalness.

4.3.2 Phonemes durations

Once each phoneme is associated to a note, their durations can be computed so that the sum of the durations of the phonemes in a note is equal to the note's duration. A default duration is given to each consonant, and the vowels' duration is then obtained by subtracting the durations of the consonants from the note's duration.

Figure 4.2 shows the distributions of the durations (in s) for each consonant, obtained from the phonetic segmentation of each database, (MS, EL and RT), and the mean values from the 3 databases.

As one can observe on those plots, each phoneme vary in a certain range, more or less large depending on the phoneme, between approximately 0.05s and 0.5s.
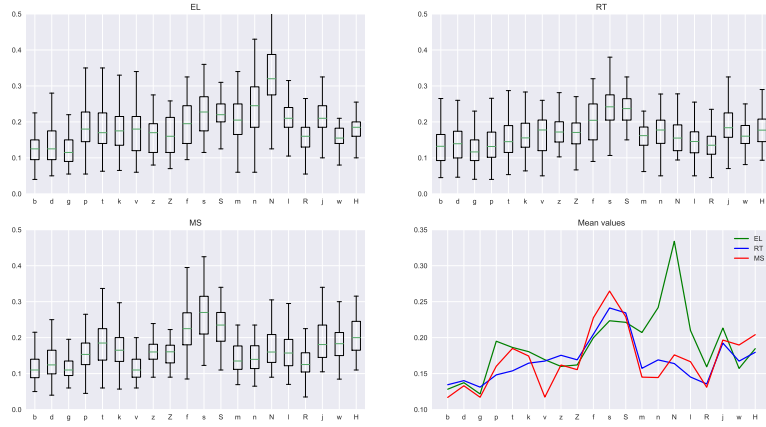
FIGURE 4.2: Distributions of consonants' durations, in seconds, for databases EL, MS and RT, and comparison between their mean durations

These values seem rather consistent across the various databases (the mean values for most phonemes are relatively similar from one database to another). However, one can also observe slight differences between singers. For instance, MS tends to use slightly longer unvoiced fricatives than EL, and EL has particularly long nasals compared to other singers. One can also observe that phonemes that share similar articulatory (and thus acoustic) similarities also have similar durations for each singer. This is the case for instance for voiced plosives (/b/, /d/, /g/), unvoiced plosives (/p/, /t/, /k/), voiced fricatives (/v/, /z/, /Z/), unvoiced fricatives (/f/, /s/, /S/), nasals (/m/, /n/), or semi-vowels (/w/, /j/, /H/).
In our system, we use as default values for the duration of each consonant the mean duration analyzed on the synthesis database, as plotted in figure 4.2. These values are observed for a regular and rather slow rate, which thus results in a natural elocution if the notes in the score are long enough, for relatively slow tempi. But for short notes, some rules should be used to adapt those durations to fit into the note's duration, while keeping a natural elocution. A simple rule is used in our system: a maximum duration is set for the group of consonants contained in each note, set as a ratio of the note's duration, so that there is always a minimal duration for the vowel. Figure 4.3 shows the distribution of the ratios of consonants groups durations over the corresponding notes' durations, analyzed on 12 songs from 4 singers with different singing styles, rhythm, and tempi. This corpus of songs will be further described in the next chapter, section 5.3. Based on this analysis, the maximum ratio has been set to 0.85.
Let's denote d_{note} a note's duration, d^i_{cons} the duration of the ith consonant contained in this note, and $d^{total}_{cons} = \sum_i d^i_{cons}$ $(i \geq 1)$ the total duration of the group of consonants contained in this same note. Then, in case $d^{total}_{cons} > 0.85 \cdot d_{note}$, the consonants' durations are uniformly compressed according to the following equation:

$$\delta = d^{total}_{cons} - 0.85 \cdot d_{note}$$
$$d^i_{cons} = d^i_{cons} - \frac{d^i_{cons}}{d^{total}_{cons}} \cdot \delta \quad (4.1)$$

This rule has been empirically chosen in order to approximate a natural behaviour
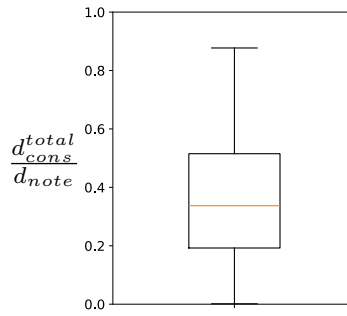
FIGURE 4.3: Distribution of total consonants durations over notes' durations, from a corpus of 12 songs from 4 singers with various styles and tempi. Consonants may occupate up to 85% of the note duration

for the pronunciation of syllables while following a given rhythm: for long notes, the mean durations from real singers at a slow speed are used without modification, and for shorter notes, the consonants are uniformly compressed in order to fit into the note while leaving some space for the vowel. Note that a similar strategy has been described in [BB16b] for generating and compressing the phonemes' durations (setting the minimum duration of vowels to 25% of the note).
But this approach is rather simplistic, and in reality, many factors like the phonetic context or expressive intentions may influence the consonants' durations even in the case of long notes where the given rule wouldn't be applied. For instance, the duration may not be the same if a consonant is surrounded by only vowels or by other consonants (independently of the note's duration), or a consonant may be purposely lengthened to accentuate a note.

Figure 4.4 represents the duration of each consonant in a song extract sung by the same singer (MS) at 3 different tempi (60, 82, and 100 BPM). The lyrics from this extract (in French), are: *"J'irai chercher ton coeur, si tu l'emportes ailleurs, même si dans tes danses, d'autres dansent tes heures."* and the consonants are presented in the figure by order of appearance in the sentence (the vowels are not represented). As one can see, the slower the tempo, the longer the durations in most cases. But the stretching ratio varies from one consonant to another, which indicates that the tempo (and thus notes' durations) is not the only factor to be considered for adapting the phonemes' durations.

Figure 4.5 shows the durations of groups of consonants contained in each note (d_{cons}^{total}) against the note duration for the same set of song as for figure 4.3. As can be observed, the consonants' durations tend to increase with the note's durations as expected, until the notes reach a certain duration (around 1s). For long notes, the consonants' duration is thus not influenced by the note's duration anymore. Moreover, one can observe that the consonants' duration is always contained below 85% of the note's duration (shown by the black line), which corresponds to the behaviour we implemented as explained above. However, in all cases, the range of possible durations is still rather large. This variability is obviously partly due to differences related to phonemes identity, but may also be related to other factors, like specific musical intentions, and this possibility should thus be further explored. We will see in next chapter 5 a more advanced way to choose the durations of
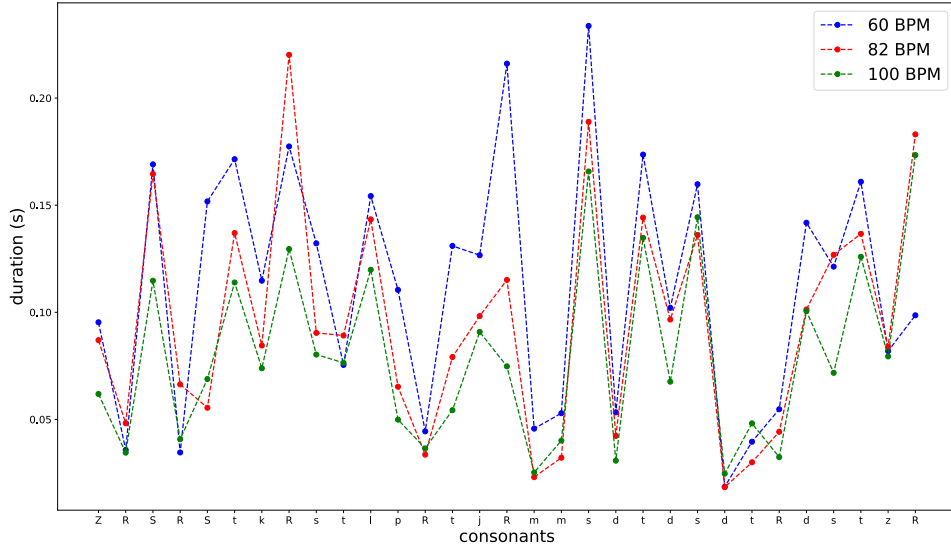
FIGURE 4.4: Durations of consonants for a song extract at 3 different tempi
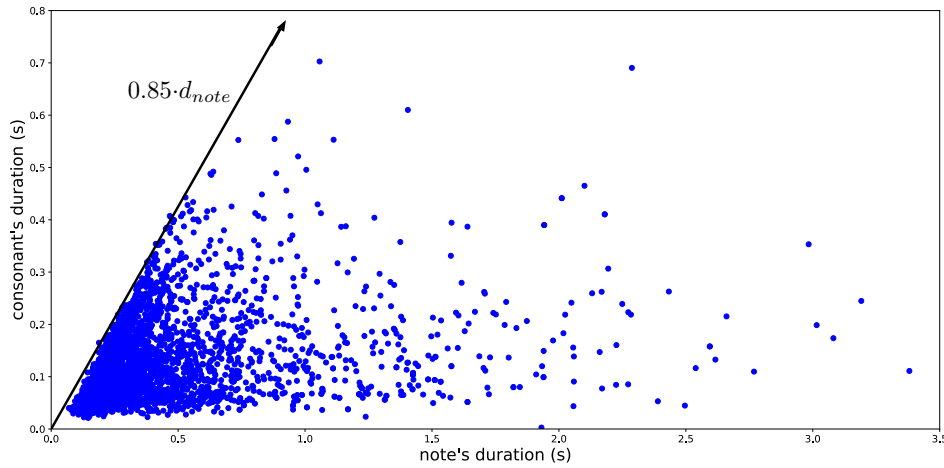
FIGURE 4.5: Consonants' durations (d_{cons}^{total}) vs. notes' durations (d_{note}) for the same set of songs as figure 4.3

consonants according to various contextual informations (including rhythmical, but also phonetic and melodic features), while modeling the singing styles of specific singers based on recordings.

4.4 f_0 modeling

Among the various control parameters, fundamental frequency (f_0) is especially important as it conveys not only the melody, but also many expressive and stylistic features, as well as some mechanical characteristics [SG09; Cha13; Kak+09; NLM07]. The main interest in using SVS softwares is to give the user a complete control over the synthesis. In particular, for artistic purposes, composers need to have control over expressive parameters of the f_0 curve, which is missing in many current approaches. f_0 models should thus have the ability to generate natural contours reproducing accurately the various f_0 fluctuations, while allowing

a flexible and intuitive control of expressivity to meet a particular style or musical idea of the composer. This need for controllability is especially more important than for speech, for which the aesthetic and stylistic characteristics have fewer importance.
As reviewed in sections 2.6.1.1 and 2.6.2, several methods have already been developed for generating f_0 curves for singing voice, among which one can mainly mention HMM-based methods [Sai+06; STK10; Our+10; Lee+12; LDL12; Umb15], 2nd order linear systems [SUA02; SUA05; Sai+07; Ohi+12], and unit-selection based models [UBB13a; Umb+15; Umb15]. Although these methods may be appropriate to synthesize natural f_0 curves while also carrying some expressive features, they don't provide means for the composer to edit the curve locally and easily modify the expressivity.
For this purpose, it would be advantageous to use a parametric model allowing to characterize expressive fluctuations of the f_0 like preparation, overshoot, or vibrato in an intuitive way. The 2nd order linear system-based method proposed in [SUA05] is parametric. However, even though the model parameters are physically meaningful, they are not from a composer point of view. In HMM-based approaches, the f_0 is parametrized by low-level features like the mean and variance of the f_0 and its derivatives (and possibly vibrato-specific parameters) over clustered context-dependant states. If those statistics may be manipulated to modify the global characteristics of f_0 (e.g. using model adaptation [Tam+01b; Shi+14] or interpolation [TD12] techniques), this approach doesn't provide any local control of the curve.
A high-level parametric model may thus provide such local control to characterize and quantify the expressive f_0 variations. Each singer and singing style has its own characteristics that should ideally be represented in a common framework. However, one may not expect to be able to precisely characterize every fine details of the f_0 contours for all singing styles and singers with such a high-level parametric model while using only a restricted set of meaningful parameters. There is thus a compromise to be found between the simplicity and controllability of the model and its generality and flexibility, to model most of the perceptually relevant expressive characteristics using a restricted set of meaningful parameters.

To achieve this goal, we thus present in the following sections a novel parametric f_0 model for singing voice synthesis, offering intuitive control of expressive parameters, mainly focusing on Western-European singing styles, which constitutes one important contribution of this thesis. The proposed approach considers the various types of f_0 fluctuations of the singing voice as separate layers, using B-splines to model the main melodic and expressive features.

4.4.1 Model overview

As exposed in 2.6.1.1, various types of f_0 fluctuations can be identified, some of which are mainly related to uncontrolled mechanical articulatory behaviours (jitter and micro-prosody), and others being mainly related to singing styles and expressivity (vibrato, preparations, overshoots, attacks, transitions' durations, ...). In [SUA05], the authors studied how those various fluctuations affect the perception of synthesized singing voices, and concluded that all types of fluctuations have importance on the perception of naturalness.
We propose here to model those variations by decomposing the f_0 curve into several additive layers:

- a melodico-expressive component, comprising the sustained notes at given pitches carrying the vibrato, attack and release parts, and transitions between notes.
- a micro-prosodic (or phonetic) component, representing the phoneme-dependant f_0 inflexions induced by the pronunciation of some consonants
- the jitter (also sometimes referred as "pitch drift" or "fine fluctuations"), that corresponds to random uncontrolled variations of the f_0.

The first layer carries the melody and most of the expressive and stylistic characteristics, while the former two are mainly related to uncontrolled mechanical behaviours which confer some naturalness to the voice.

Each of those layers is thus modeled independently, and the f_0 curve is then obtained as a simple summation, as shown on figure 4.6. Some similarities may be found with superpositional models used in speech synthesis which differentiate for instance f_0 variations at the sentence's level from the more local accent and phoneme's levels to model independently the different components of voice prosody [FH84; SMK04; Sak05].
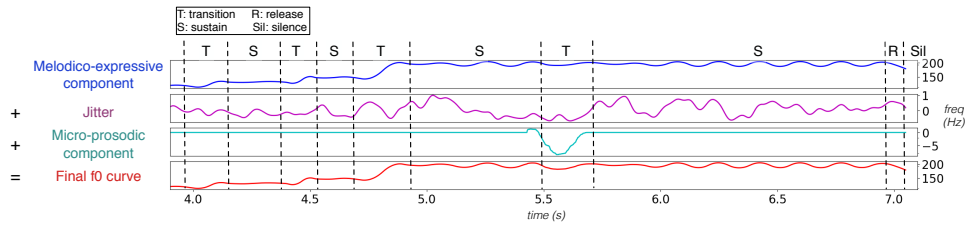
FIGURE 4.6: Vertical and temporal decomposition of the f_0 curve. The 3 layers are modeled independently and add up to form the final target curve.

In a first version of our model, presented in [ADR15], the vibrato was modeled as a separate layer. However, the model has since then been slightly modified to integrate some improvements, and the melodic and vibrato components have been merged into a single "melodico-expressive" component, as will be explained below.

In addition to this “vertical” decomposition in multiple layers, we also define an “horizontal” decomposition to model the evolution of the f_0 across time according to the input score. From this temporal point of view, we model the f_0 curve as a succession of 5 basic types of segments: attacks, sustains, transitions, releases, and silences (in a similar way to what is done in [MBL06]), as shown on figures 4.6 and 4.7. This temporal segmentation applies to the melodico-expressive component, which models each of those segments in a parametric way using B-splines, such as will be described in the next section. Figure 4.7 shows such horizontal decomposition, along with the various identified types of f_0 variations on a real singing recording. Note that transitions containing unvoiced phonemes are treated similarly to voiced transitions (with a continuous curve) in our model, which is conceptually simpler and consistent with the notion of transition. Although the f_0 is not visible during the unvoiced part, the transition may still contain a voiced part that exhibits fluctuations similar to voiced transitions, and it thus makes sense to
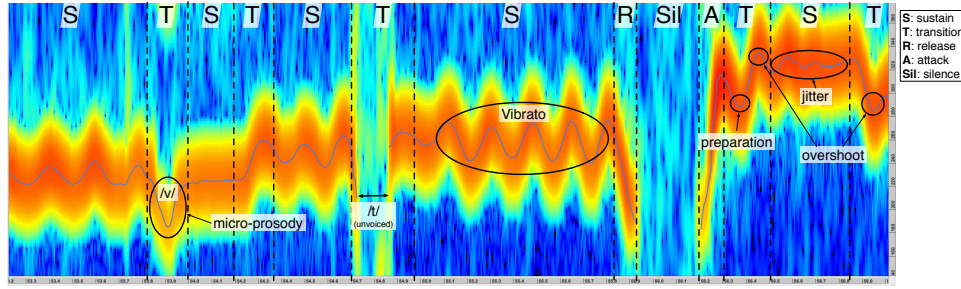
FIGURE 4.7: Extract of a real f_0 curve with its horizontal decomposition, showing the various types of possible fluctuations

model them similarly.

The following sections detail the proposed model, with the approaches used for generating each layer.

4.4.2 Melodico-expressive component

The melodico-expressive component constitutes the main layer, that carries most of the expressive features of the f_0. As explained before, this layer is segmented, in our model, into 5 elementary segments, which are: silences, attacks, sustains, transitions, and releases. For providing an intuitive control of expression to the user, each of those segments is parametrized with a restricted set of meaningful parameters.

4.4.2.1 Parametrization of the curve

Figure 4.8 summarizes the various control parameters of the proposed model, related to each type of segment, that may be used to shape the curve and thus control the expressivity of the voice.

Attacks/Releases:
Attacks are characterized by a rising slope, at the beginning of a sentence, after a silence. Symmetrically, releases are constituted by a descending slope of the f_0, at the end of a sentence, before a silence segment. Both are parametrized by their length L (in seconds), and their depth D (in cents: $D = 1200 \cdot \log_2(\frac{f_{min}}{f_{end}})$), similarly to glissup and glissdown segments in [IIO14b].

Transitions:
Transitions are smooth continuous f_0 segments joining 2 successive notes together (possibly interrupted during unvoiced phonemes). As pointed out in [SUA05], transitions may carry 2 types of fluctuations that are important to the perception of singing voice, which are preparations and overshoots. Overshoots are a type of inflections characterized by the f_0 exceeding the target note frequency for a short amount of time at the end of a transition. Preparations corresponds to similar kinds of deflections in the opposite direction of the pitch variation between 2 notes, at the beginning of a transition.
We thus built a transition model that allows such fluctuations, as described in figure 4.8. Transitions may be asymmetrical and are thus split into 2 parts around the center, shaped by a total of 4 parameters. The lengths of the left and right parts are
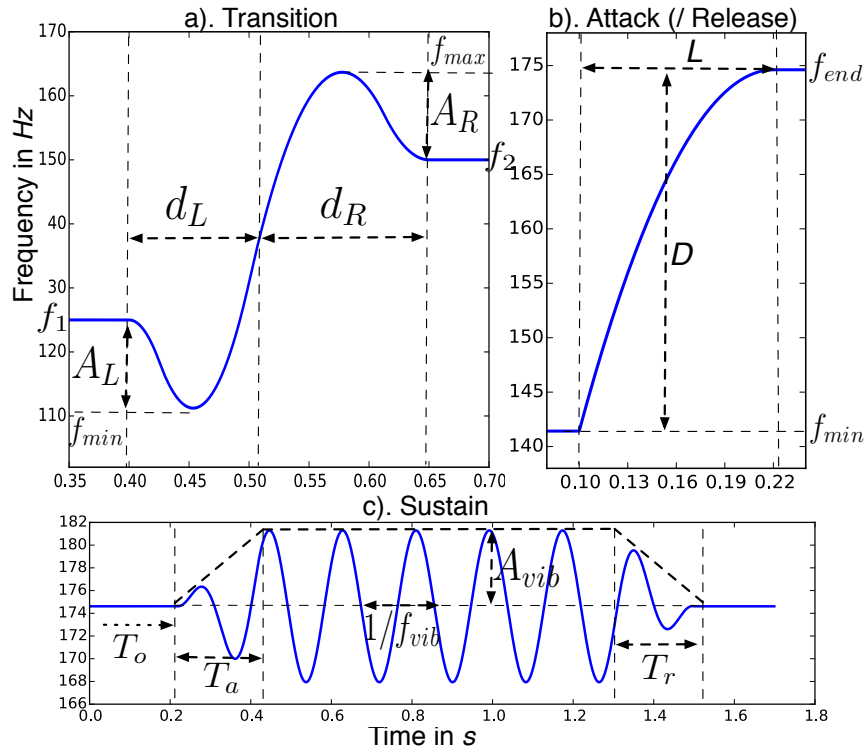
FIGURE 4.8: f_0 model parameters

determined by the parameters d_L and d_R respectively (in seconds). The amplitudes of the preparation and overshoot are given by the parameters A_L and A_R (in cents: $A_L = 1200 \cdot \log_2(\frac{f_{min}}{f_1})$ and $A_R = 1200 \cdot \log_2(\frac{f_{max}}{f_2})$). For an upward transition, we have $A_L \leq 0$ and $A_R \geq 0$, and the inverse for downward transitions. Figure 4.9 shows various possible transitions' shapes that may be obtained using different sets of values for those 4 parameters. Note that a transition between 2 notes at the same pitch, can also contain a downward inflection that can be modeled similarly to an upward transition, using a negative value for A_L, and $A_R = 0$, as shown in figure 4.9 f). Corresponding sounds for the transitions examples in figure 4.9 are also attached (sounds 4.2 to 4.7).

Sustains:
Sustains constitute the stable parts of notes that support the target pitch of the note and that often carry some vibrato. The vibrato is characterized by its frequency f_{vib} (in Hz), and an amplitude curve of type Attack-Sustain-Release (ASR), shown in figure 4.8 (dashed lines). This ASR curve is determined by a global amplitude parameter A_{vib} (in cents), an attack time T_a, and a release time T_r (in seconds). Additionally, an offset time T_o allows to set a delay between the start of the sustain segment and the start of the vibrato.

The proposed parameters are directly related to expressive fluctuations of the f_0 in terms of duration and frequency, and can thus be easily manipulated for reshaping the curve.
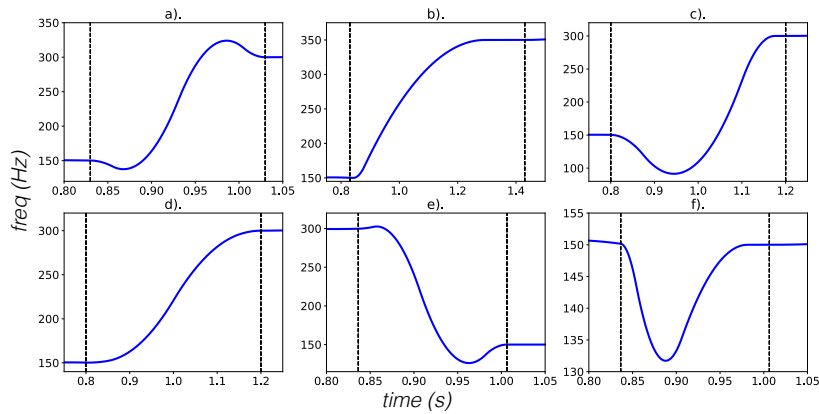
FIGURE 4.9: Examples of possible transitions shapes using different parameters' values, for upward (a,b,c,d), downward (e), or same-note (f) transitions.

4.4.2.2 Mathematical background: B-splines

For generating such parametric f_0 curves, some mathematical model is necessary. Several possible approaches have been considered for this purpose in the literature. We already introduced in 2.6.1.1 the approach used in [SUA02; SUA05; Sai+07], based on 2^{nd} order linear systems' equations to generate the vibrato, preparations and overshoots.
The Discrete Cosine Transform (DCT) has been used in several studies for characterizing f_0 curves for speech and singing [TWR08; LA08; SG11; Dev+11], using the first coefficients to capture the mean, slope, and curvature of f_0 segments at various levels.
In [Bat04] and [MBM06], authors consider the use of Bezier curves for fitting and characterizing singing voices' f_0 curves, outlining some potential applications in the field of singing voice synthesis or transformation and computer-assisted composition.
However, concerning the DCT and Beziers curves-based approaches, it seems that no full model allowing the generation of f_0 curves from a score have been implemented.
Furthermore, while those 3 approaches remain parametric, the parameters provided to shape the curve are not intuitive to control for a user, as not directly related to perceptually-relevant features.
In the same direction than approaches based on Bézier curves, the authors in [BBL05; Lol06; LBB10] proposed to fit speech f_0 curves using B-splines, with a high accuracy. The authors in those studies don't propose neither a model for the generation of f_0 curves for speech or singing synthesis. However, those studies show the potential of B-splines, as a mathematical tool, for fitting, and thus modeling real f_0 contours, which suggests that B-splines might be a good candidate for generating smooth and expressive curves in the context of singing voice synthesis. A particular advantage of B-splines compared to Bezier curves and DCT is that it offers a better local control of the curve's shape. For this reason, we thus chose to use B-splines as the mathematical basis underlying our f_0 model for the generation of the melodico-expressive component. In the following, we will first shortly introduce the necessary mathematical background related to B-splines. Then, we will explain how we use it in our model for generating the f_0 fluctuations

inherent to each segment according to the parameters shown in figure 4.8.

While approaches like the DCT decompose a contour on a set of functions that are globally defined on a temporal segment, B-splines allow decomposing a contour over a set of overlapping functions that have non-zero values only on a local segment.
A B-spline function denoted $B_m^i(t)$ is a piecewise polynomial function using polynomials of degree m, with non-zero values only on a segment i. Considering a vector of increasing real values $(\hat{t}) = (t_0, ..., t_k)$ called "knots", such a function is defined recursively as follows:

$$B_0^i(t) = \begin{cases} 1 & \text{if} \quad t \in [t_i, t_{i+1}[\\ 0 & \text{otherwise} \end{cases}$$
$$B_m^i(t) = \frac{t-t_i}{t_{i+m}-t_i} B_{m-1}^i(t) + \frac{t_{i+m+1}-t}{t_{i+m+1}-t_{i+1}} B_{m-1}^{i+1}(t) \tag{4.2}$$

where, by convention, fractions equal to zero in case $t_i = t_{i+m}$ or $t_{i+m+1} = t_{i+1}$. Note that values in $(\hat{t})$ can thus be repeated several times, such that $t_{i+1} = t_i$.

From this formulation, each B-spline function B_m^i is such that

$$B_m^i(t) \begin{cases} > 0 & \text{if} \quad t \in [t_i, t_{i+m+1}[\\ = 0 & \text{otherwise} \end{cases} \tag{4.3}$$

Let us define an interval $[a, b]$ segmented into $l + 1$ sub-intervals using knots t_i: $a = t_m < ... < t_{m+l+1} = b$. Setting $t_0 = ... = t_m = a$ et $t_{m+l+1} = ... = t_{2m+l+1} = b$, the set of B-splines functions defined by the knots vector $(\hat{t})$ form a basis of a vector space of dimension $m + l + 1$.
Moreover, so-defined B-splines functions satisfy the condition

$$\forall t \in [a, b[, \quad \sum_{i=0}^{m+l} B_m^i(t) = 1 \tag{4.4}$$

The value $m + 1$ is called the order of the B-splines. A basis of overlapping 3rd order B-spline functions is shown in figure 4.10, along with their summation (equal to 1 at all positions) and the knots' positions.
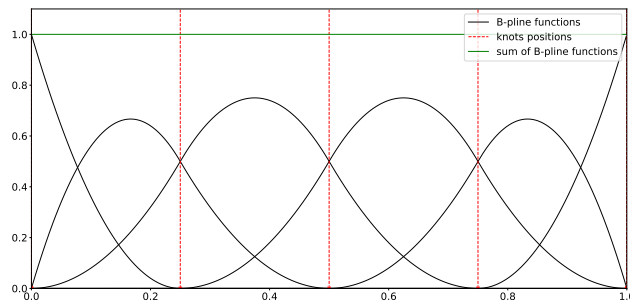
FIGURE 4.10: A basis of 3rd order B-spline functions defined on time segment $[0, 1]$

A smooth continuous B-spline curve spanning the time segment $[a, b[$ can then be obtained as a linear combination of the B-spline functions B_m^i of this basis:

$$f(t) = \sum_{i=0}^{m+l} c_i B_m^i(t) \tag{4.5}$$

where the weights c_i are usually called the control points. Using this approach, changing the value of the coefficient c_i will only affect the B-spline B_m^i, which is 0 outside of the interval $[t_i, t_{i+m+1}[$. This thus allows a very local control of the curve's shape, which is an advantage for our purpose.

Another property of such B-spline curves is that they belong to the category of C^{m-1} continuous functions, which means that the function and its $m-1$ first derivatives are continuous. It is thus possible to generate a smooth f_0 curve using B-splines by setting appropriate knots and control points. Note that the continuity property can however be broken by setting several knots to the same value t_i. If $t_i = t_{i+1}$, then the curve is only C^{m-2} continuous at time t_i. If $t_i = t_{i+1} = t_{i+2}$, it is C^{m-3} continuous at time t_i, and so on ...

4.4.2.3 Curve generation

In order to generate the f_0 curve automatically from the score, a sequence of the model's segments must first be determined from the notes, and their parameters be chosen. Then, the knots vector $(\hat{t})$ and control points (or weights) c_i are determined according to those parameters. Finally, a B-splines basis is built from the knots vectors and the curve is obtained, following equation 4.5. In our model, we use 3rd order B-splines. The steps of this process for generating the f_0 curve from the score and given parameters are illustrated in figure 4.11. For the sake of simplicity and clarity, the vibrato does not appear in this figure, as it will be the subject of a next dedicated section.

The sequence of segments is determined as follows, according to the notes durations and the model's temporal parameters (d_L, d_R, and L): silence segments are given by the rests in the score; an attack segment is always placed just after a silence, and a release just before; a transition is placed between each pair of notes; a sustain segment is positioned for each note, in-between the surrounding transitions and/or attack and release segments.
While some authors assume that the pitch change in transitions should be completed by the time of the vowel's onset [Ber96; Mac+97b; Fon01; Bon+01a; Sun06], we found out from the observation of recordings that this is not always the case, and that transitions can often span the 2 notes. In our system, transitions are thus by default centered on the notes' onsets (i.e. the vowels' onsets). Optionally, an additional offset parameter δ_t may be used to shift the transition around the note's onset. However, when transitions contain consonants, some specific rules are used to most appropriately position the transitions coherently with the lyrics, as will be explained in section 4.4.2.6. Then, the start and end times of the transitions are positioned according to the parameters d_L and d_R. The attacks begin on the note's onset, and the releases end at the note's offset. The respective end and start times are set according to parameter L. Some additional rules for the positioning of attacks and releases will also be detailed in section 4.4.2.6. The times of the sustain segments are determined afterwards so to fill the remaining gaps between
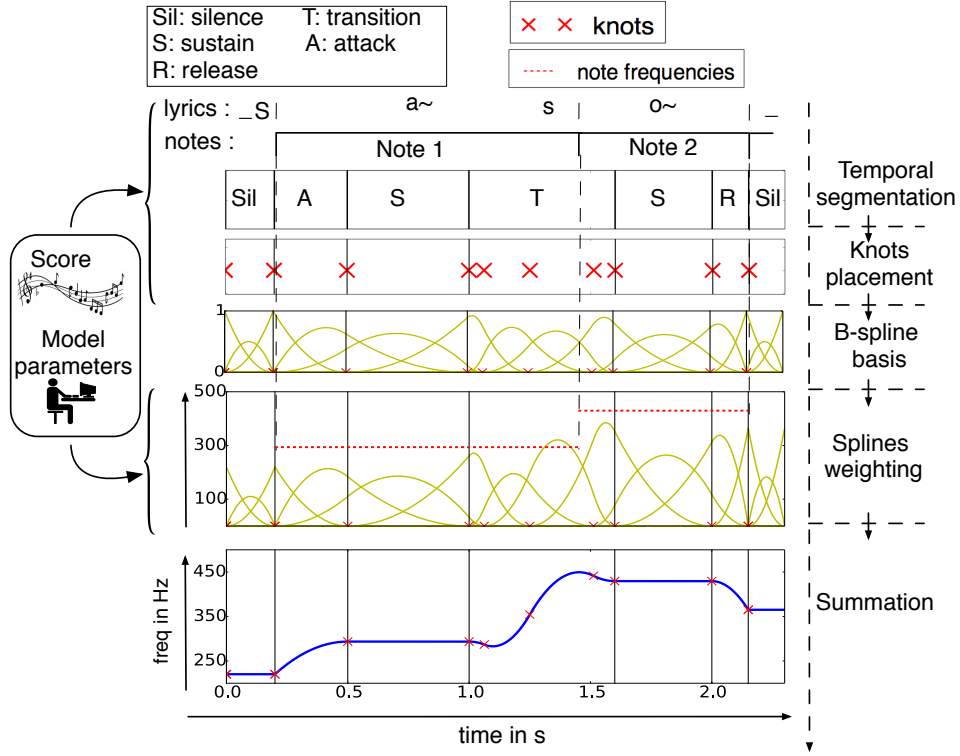
FIGURE 4.11: Process of generating the melodico-expressive component using B-splines.

other segments during sustained notes, once all other segments have been placed appropriately.

Then, each segment is shaped using a specific number of B-splines functions. After a first automatic pass using default parameters for all segments, the sequence of segments and their control parameters may be modified by the user to change the expression (from a description file in the xml format). In the next chapter 5, we will detail an approach we developed to automatically choose the parameters of each segment individually, according to local musical contexts, while trying to model specific singing styles. We will now explain how the knots' positions and the splines' weights are determined according to the parameters of each segment.

For transitions, 5 knots are positioned as described in figure 4.12. The 1st, 3rd and 5th knots are placed at the start, middle and end times of the transition. The 2nd and 4th knots are placed at $0.75 \cdot d_L$ and $0.75 \cdot d_R$ from the middle knot. These knots' positions then serve to generate a set of 3rd order B-spline functions, which are then weighted in order to shape the transition, as illustrated in the figure. The weights' values are determined as follows, from the model parameters:

$$
\begin{aligned}
w_1 &= f_1 \\
w_2 = f_1 + 2 \cdot \delta_{f_{A_L}} \quad &\text{with} \quad \delta_{f_{A_L}} = 2^{\frac{A_L}{1200}} \cdot f_1 - f_1 \\
w_3 = f_2 + 2 \cdot \delta_{f_{A_R}} \quad &\text{with} \quad \delta_{f_{A_R}} = 2^{\frac{A_R}{1200}} \cdot f_2 - f_2 \\
w_4 &= f_2
\end{aligned}
\tag{4.6}
$$

where f_1 is the target frequency of the left-side note, f_2 is the target frequency of
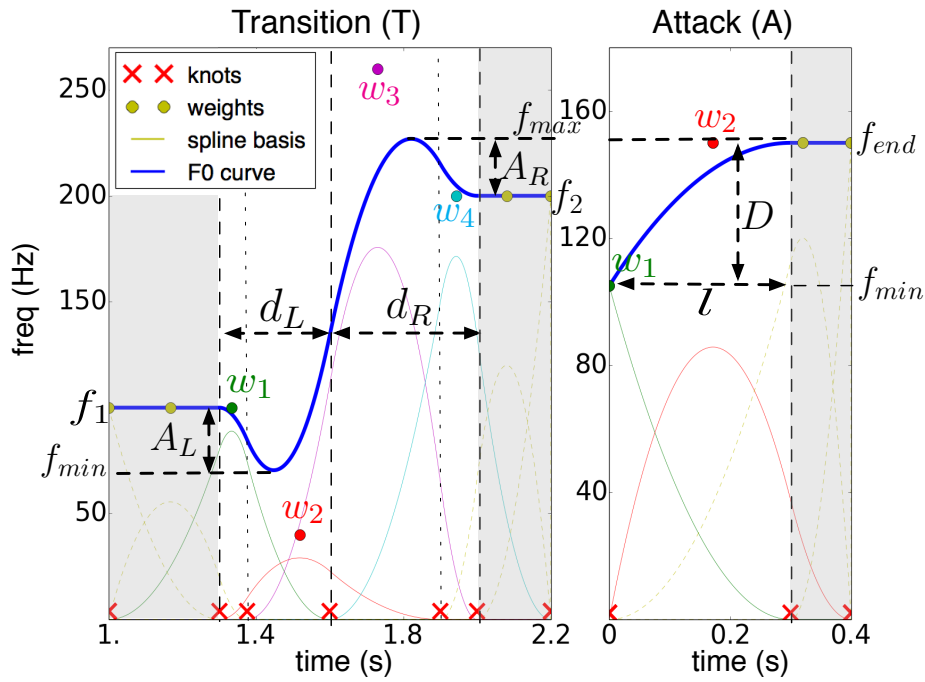
FIGURE 4.12: Transition and attack/release models, showing underlying B-splines, along with knots positions and weights

the right-side note, and $\delta_{f_{A_L}}$ and $\delta_{f_{A_R}}$ are the values corresponding to A_L and A_R (initially given in cents) converted to frequencies.

These weights' values have been empirically chosen to approximate at best the target parameters. But because the B-splines functions are overlapping, the resulting curve may not exactly match the values of A_L and A_R, as the curve value at one point depends on several weights (and thus the value of A_R may slightly influence the real depth of the preparation in transitions for instance). Obtaining an exact results would imply to take into account in the computation of each weight w_i the influences of the weights of the surrounding overlapping B-splines (from w_{i-2} to w_{i+2}), which would thus be much more complex. Note however that the values of those weights have been revised since the first publication of the model in [ADR15] to better match the target values A_L and A_R, and the formulas given in equation 4.6 give satisfying results, very close to the target values in most cases. However, for a better precision in some cases, a correction of the curve has been introduced, as will be explained in section 4.4.2.5.

The curve is finally generated by summing all the weighted B-spline functions along the time axis. While conceiving this model, 3-knots and 7-knots models had also been considered for generating transitions. However, using only 3 knots doesn't provide enough flexibility for producing adequately f_0 fluctuations like preparations and overshoots. Contrarily, a 7-knots model gives more flexibility to accurately model all the specific variations used by some singers during transitions, but requires more parameters to control, which becomes too heavy for the user to handle. A 5-knots model such as presented here thus seems to be a good compromise between flexibility and controllability.

Attacks and releases are defined in a similar way, as shown in figure 4.12. Knots are placed at the start and end times of attacks, and the weights of the spline vectors are:

$$\begin{aligned} w_1 &= 2^{\frac{D}{1200}} \cdot f_{end} \\ w_2 &= f_{end} \end{aligned} \tag{4.7}$$

where f_{end} (in Hz) is the target frequency of the attacked note, and D is defined in cents. Releases are modeled symmetrically to attacks.
Usually, the value of parameter D should always be negative. However, it may possibly be set to a positive value for releases in order to simulate for instance a rise of the f_0 that is typical of some Eastern-European (balkanic) singing styles, as illustrated by sound 4.1.

Our system currently doesn't provide a GUI for managing the parameters which can only be modified manually in an xml file. Nevertheless, one may easily imagine a convenient interface where the user could shape the transitions by moving 3 handles in a 2D time-frequency space: one to control the transition center (δ_t), one 2D handle for (d_L, A_L), and another 2D handle for (d_R, A_R). Similarly to transitions, one may imagine a convenient interface providing a single 2D handle to control (L, D) for attacks and releases.

So far, we have presented how we can generate a smooth baseline f_0 curve carrying various expressive fluctuations such as attacks, releases, preparations and overshoots, but without vibrato. We present in the next section how the vibrato is generated in our model for the sustain segments.

4.4.2.4 Vibrato generation

As already outlined in section 2.6.1.1, vibrato is one of the most important expressive features of singing voice, as it relates to singing style, singer's individuality and proficiency, and is especially important for some styles like lyrical singing. It is characterized by a quasi-periodic modulation of the f_0 at a rate usually lying in a range from 5. to 7.5Hz and an amplitude from ± 0.5 up to ± 2 semitones.
While some authors have seeked to precisely characterize the vibrato shape, rate and amplitude [STK10; IIO14a], the necessity of such a precise description of the vibrato for synthesis purpose, from a perceptual point of view, has not been attested. [MB90] suggests that a good vibrato is nearly sinusoidal and that changes in its shape along time is not perceived by listeners. In [Sun06], the author studies the differences of f_0 fluctuations between 2 singers perceived respectively as being good and bad singers, and concludes that one of the aspects conferring its bad quality to the 2nd voice is a particularly irregular vibrato. These assumptions thus encouraged us to use in our system a very simple vibrato model, consisting in a sinusoid with a fixed frequency f_{vib}, and a simple ASR amplitude curve, as presented previously (see figure 4.8), rather similar to the model presented in [SF01] for violin vibrato.

In the first version of our model, presented in [ADR15], the vibrato was generated as a separate layer using a simple sinusoïd. However, the problem of this first approach, using a fixed frequency, is that it didn't allow to properly synchronize the first and last vibrato cycles to smoothly connect the vibrato with the overshoot and preparations of the surrounding transitions. In some cases, it could lead to unusual fluctuations at the borders of the sustains segments which could sound unnatural.
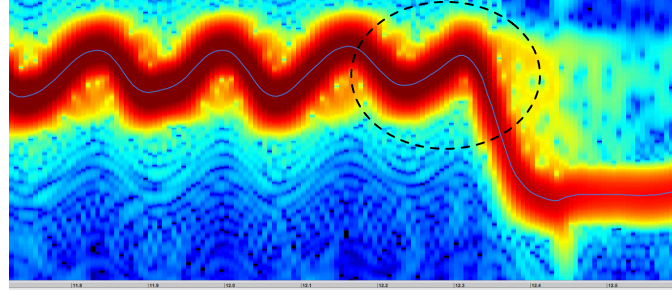
FIGURE 4.13: Example of a transition smoothly chained to a vibrato, from a recording. The preparation is in phase with the vibrato.
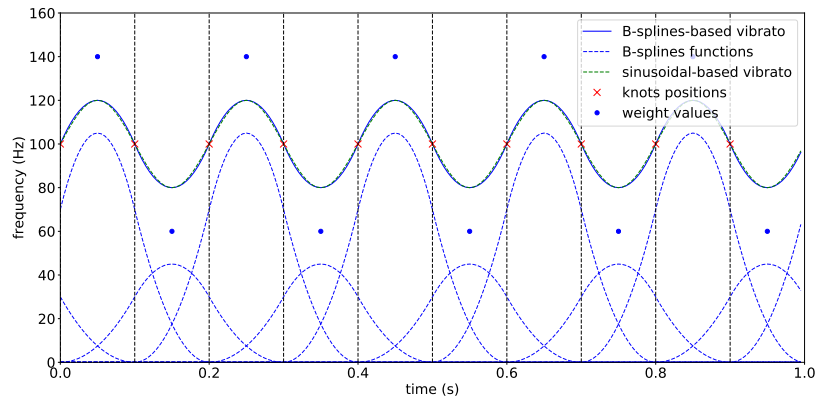
FIGURE 4.14: Comparison between a 5Hz sinusoidal vibrato (dashed green) and B-splines-generated vibrato (solid blue), along with knots positions, weights, and underlying B-splines functions.

Such problems are studied and illustrated in [MM08].
For overcoming this problem, our solution is to also generate the vibrato using B-splines on sustain segments, so that all expressive fluctuations are generated in a unified framework, all included in a single melodico-expressive layer, and can be smoothly connected thanks to the continuity properties of B-spline curves. This approach makes more sense, as in case the amplitude of the overshoots or preparations match that of the vibrato and the attack and release times are set to 0, the preparations and/or overshoots naturally merge with the vibrato, as shown in figure 4.13 where the 2 are properly in phase.

In order to generate the vibrato using B-splines, we simply position knots during sustains between each half vibrato cycle and alternate the weights values around the nominal pitch of the note, as shown in figure 4.14. The values of the weights for generating such a vibrato are set as follows:

$$w_i = f_0 + a_i \cdot 2 \cdot (2^{\frac{A_{vib}(i)}{1200}} f_0 - f_0) \tag{4.8}$$

where f_0 is the frequency of the note, $A_{vib}(i)$ is the amplitude, in cents, of the vibrato at the position i (center of the i^{th} half cycle) obtained from the ASR amplitude curve shown in figure 4.8, and a_i is a coefficient alternating for each weight (and thus each half-cycle) between values 1 and -1.
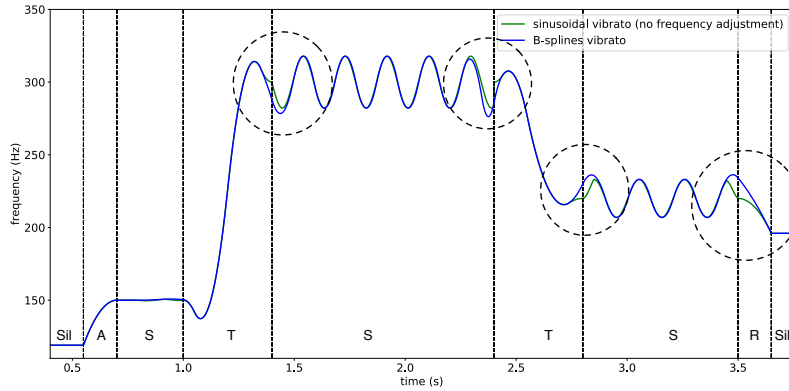
FIGURE 4.15: Comparison between sinusoidal and B-splined based vibrato at the junctions with transitions with preparations and overshoots.

As one can see on the figure, the so-generated B-spline vibrato is almost equal to the sinusoidal one, the difference being imperceptible, which makes this approach perfectly suitable for generating natural-sounding vibratos, the weights used perfectly matching the given target vibrato's amplitude.
The value of a_0, for the first half-cycle of the vibrato, is determined according to the direction of the preceding transition. If the note carrying the vibrato is preceded by a lower note, and thus an upward transition, the overshoot is positive, and the first half vibrato cycle should thus go in the opposite direction in order for the phase of the transition and the vibrato to be coherent, which means that we need to set $a_0 = -1$. Conversely, if the vibrato is preceded by a downward transition, the overshoot is negative, and we would thus have $a_0 = 1$.
Similarly, at the end the vibrato, the phase should also be coherent with the following transition, depending on its direction. However, depending on the vibrato frequency f_{vib} and the duration of the sustained segment d between the transitions, the number of half-cycles $n = 2 \cdot f_{vib} \cdot d$ produced during the sustained portion of the note may not give the required phase at the end of the vibrato. In such case, the vibrato frequency is slightly adjusted so that so that the integer number of half cycles is coherent with the direction of the following transition, and such that the duration of the last half cycle is $> \frac{1}{4f_{vib}}$ to avoid having a too fast unnatural fluctuation at the end of the vibrato. With this approach, the last half cycle of vibrato might be shorter than others, but this is coherent with observations made in [Pra94; BS02], stating that the vibrato rate tends to increase at the end of notes. If the sustain segment is too short ($d < \frac{1}{2f_{ivb}}$), no vibrato at all is applied. Figure 4.15 shows an example of a generated f_0 curve comparing the 2 approaches to vibrato generation. As can be seen the junctions between the transitions (and release) and the vibrato are smoother using the B-splines-based approach.

4.4.2.5 Correction of the curve

As evoked previously, the values used for weighting the B-splines (detailed in section 4.4.2.3) are appropriate in most case to match the target expressive parameters. But in a few cases (mostly in the case of important preparations A_L in upward transitions), the curve obtained using those weights may exceed the target values, resulting in a too big inflection. In order to better match the given target parameters
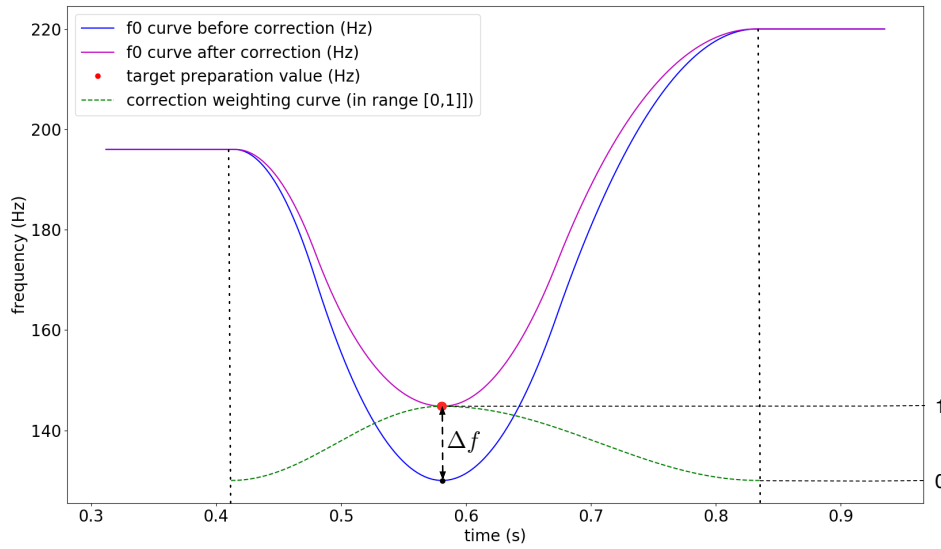
FIGURE 4.16: Correction of the f_0 curve in case of a big preparation exceeding the target value

in such cases, a correction of the f_0 curve is applied, once the melodico-expressive component has been generated from the B-splines. An example of this correction is illustrated in figure 4.16 for an upward transition with a big preparation. In this case, the difference Δf between the minimum of the curve and the target value (red dot) is first computed. Then, a correction weighting curve $w_{cor}(t)$ (dashed green line) is built, made of 2 semi-hanning windows (with values in the interval $[0, 1]$) centered on the minimum, and bounded by the curve's maxima respectively on the left and right side of the preparation. Then, the corrected curve $f_{0_{cor}}$ is obtained following equation 4.9:

$$f_{0_{cor}} = f_0 + \Delta_f \cdot w_{cor}(t) \tag{4.9}$$

Note that a similar process can also be applied to correct other inflections like overshoots, but this is most of the time not necessary.

4.4.2.6 Rules for alignment of f_0 segments to phonemes

In order to improve the coherence between the f_0 fluctuations and the timing of the lyrics, a few specific rules have been developed regarding the positions of f_0 segments, which are detailed below.

Transitions:

For transitions between 2 vowels (no consonants present), the transition is centered by default on the vowel's onset. As mentioned previously, a parameter δ_t allows to shift the transition around this point.

For transitions containing 1 or several consonants, preparations and overshoots are usually merged with the phonetic inflections of the micro-prosodic layer due to the pronunciation of certain consonants. Several cases are considered, depending on the direction of the transition:

- Upward transitions start on the first non-semi-vowel consonant. If there is only a semi-vowel, the transition starts on the semi-vowel. The transition should end on the vowel's onset or later.
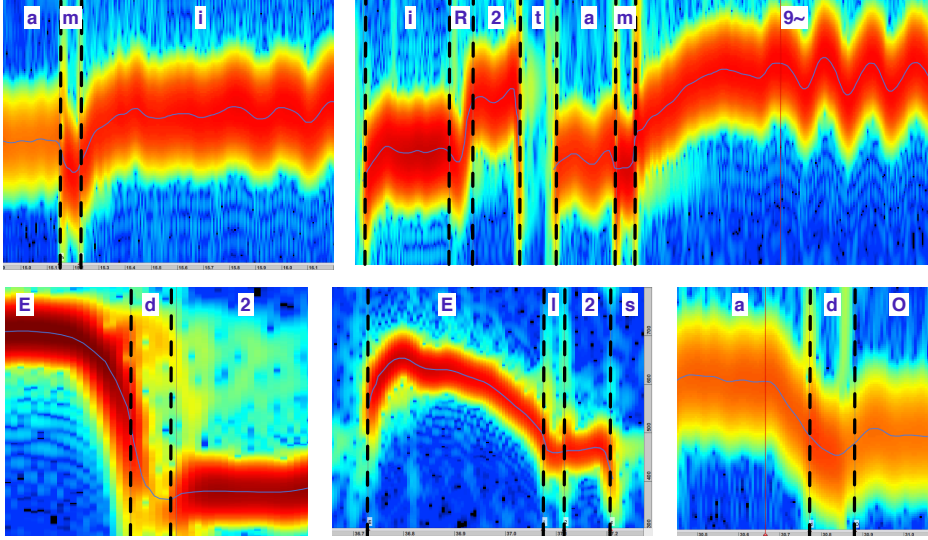
FIGURE 4.17: Examples of real f_0 contours that verify the given rule for aligning transitions to phonemes

- Downward (and same-note) transitions end on the vowel's onset, or on the preceding semi-vowel if any. If there is only a semi-vowel, the transition ends on the vowel onset. The transition should start at the first non-semi-vowel consonant or before.

Additionally, the transition can't start or end respectively beyond the start of the left vowel and end of the right vowel. The times of each transition are thus corrected according to these rules.
Figure 4.17 shows some real singing extracts along with phonetic segmentations, where such rules are verified.

Attacks:
Attacks are positioned to start at time $t_a = max((t_v - L), t_c)$, where t_v is the onset time of the vowel, L is the attack's duration, and t_c is the time of the first voiced consonant of the syllable after the silence if any (otherwise, $t_a = t_v$). Then, the attack ends at time $t_a + L$. In case the first voiced consonant of the syllable is a semi-vowel, the attack starts on the semi-vowel.

Releases:
Releases are simply placed to end at the end of the note (whether it ends with a vowel or consonants).

These rules where determined empirically from the observation of many recordings. Although they may not be always verified in real recordings, they have been found to give satisfying results in most cases, while avoiding some unnatural placements of transitions obtained sometimes when considering only the model's parameters.

4.4.2.7 Specific segments' sequences

In case the two transitions between 3 successive notes are too long (according to the given durations parameters) so that they overlap on the middle note, then the sustain segment in-between is deleted, and the times of the transitions are adjusted such
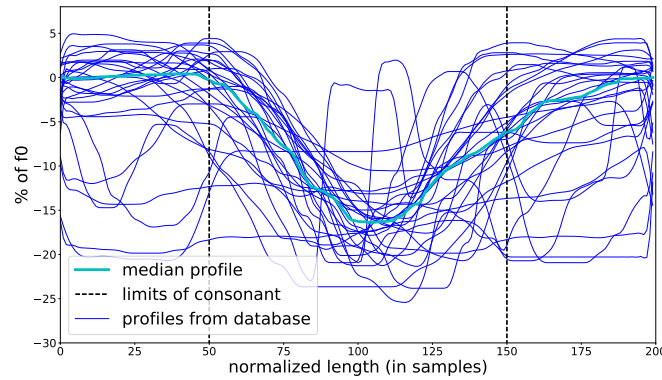
FIGURE 4.18: Example of a normalized median f_0 profile for the phoneme /Z/. The inflection spreads beyond the limit of the phoneme.

that the left transition ends when the right transition starts. The same procedure is also followed in case of a transition overlapping with an attack or release segment. Additionally, in such case, the B-splines underlying w_1 and/or w_4 (as shown in figure 4.12) are removed so that the segments can chain smoothly.

4.4.3 Micro-prosodic component

As reported by many authors (e.g. [STK10; Umb+15; BB16b; MBM06]), the pronunciation of voiced consonants induces some inflections in the observed f_0 contours. This is especially the case for voiced fricatives (/v/, /z/, /Z/) and voiced plosives (/b/, /d/, /g/). As these inflections are inherent to the pronunciation of the phonemes, they are not controlled by the singer. Thus, we decided to treat this component using an f_0 profile's template for each voiced consonant.
For this purpose, we analyzed the f_0 profiles of all occurrences of each voiced consonants in our singer database, and computed median templates for each of them. As the inflexions are usually not fully contained inside the limits of the consonant, the limits of the profiles are considered from half the consonant's length before its beginning to half its length after the end of the consonant. All extracted profiles are normalized in time (by resampling on 200 points) and frequency (as % of baseline f_0) before computing the median template. The template is then scaled during synthesis to the target length of the consonant and the target frequency given by the melodico-expressive component. Figure 4.18 shows such a template, for the phoneme /Z/.

4.4.4 Jitter component

As evoked previously, vibrato can be used to add expressivity during sustained notes. However, vibrato is not always present, and the f_0 curve is nevertheless never perfectly flat. A very flat f_0 curve is likely to be perceived as robotic or artificial. Jitter (also referred to as pitch drift or fine fluctuations) designates uncontrolled random fluctuations of the f_0. The perception of such fluctuations in the context of singing synthesis has been studied in [AK00] and [Sta11], and those studies suggest that the inclusion of such fluctuations is important for conferring naturalness to synthesized singing voices. Some artificially-generated quasi-random pitch
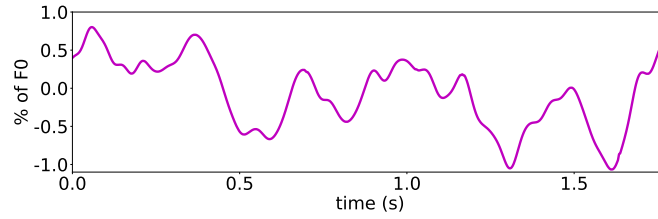
FIGURE 4.19: Example of a jitter template.

fluctuations are also induced for singing synthesis in [Mac+97b].
During an internship before the start of this thesis [Ard13], some informal tests were conducted using a generative model of jitter such as the one used in [Mac+97b], using a sum of sinusoids with various frequencies and amplitudes, or using low-pass filtered noise as in [Sta11]. However, those tests were not very conclusive. As those type of variations are inherent to the naturalness of the voice and don't need to be controlled, we thus rather preferred to use a template-based approach to generate such fine fluctuations, based on real extracted f_0 curves, similarly to the phonetic component.

For this purpose, we analyzed the f_0 curves of multiple sustained notes without vibrato contained in our synthesis databases. We normalized them in frequency according to the median frequency of the segment, and stored them as jitter templates, similarly to what is done in [Bon08a] with sustain templates. Figure 4.19 shows such a jitter template, stored as a % of the median f_0 along time.
For synthesis, we then concatenate randomly chosen templates with 200ms cross-fades at junction, for the whole length of the synthesized extract, and scale the resulting curve according to the frequency given by the baseline melodico-expressive component.

Finally, the 3 f_0 curve's components (melodico-expressive, phonetic, and jitter components) are layered and summed together to produce the final target f_0 curve to be used for the synthesis.

4.4.5 Evaluation

In order to validate the proposed f_0 model, we conducted a 3 parts listening test aiming at evaluating the perceived naturalness of the generated curves. In the first 2 parts of this test, we evaluated the relevance of the different layers, independently of each other. Then, in a 3^{rd} part, we confronted the complete model to real f_0 curves from professional singers. The test was conducted on 46 participants listening with headphones or earphones, through a web interface, using a CMOS preference test to compare pairs of synthesized singing voice extracts, as described in [Rec03]. All the sounds were synthesized using our ISiS concatenative synthesizer with the SVP engine described in chapter 3. For all parts of the test, in order to avoid any bias due to other parameters in the evaluation, only the f_0 curve was different between the 2 synthesized files of a pair, all other sound's characteristics (durations, spectral envelopes, ...) remaining the same. For each test, similar examples were synthesized using both a man and a woman voice, using the MS and RT databases (the only 2 available at the time of this study). Nevertheless, using only those 2 voices in our test does not seem critical here, since we do not aim at evaluating the overall quality of the synthesis, or timbre characteristics, but only the f_0 model. Note that the first version of the f_0 model presented in [ADR15] was used in this

test, with the vibrato being thus generated as a separate layer. Only the naturalness of the model is evaluated here, assuming that its controllability is ensured by the proposed approach. Each part of the test is detailed in the subsections below, and the results are shown in figure 4.21, with the confidence interval of 95% [Rec03]. All sounds used in the test can be found in a web page at url [2]. Figure 4.20 shows the layout of the custom web interface used for those tests (developed in php and html5).

Pair	File1	+3	+2	+1	0	+1	+2	+3	File2	Prob
1	0:00 / 0:04								0:00 / 0:04	
2	0:00 / 0:04								0:00 / 0:04	
3	0:00 / 0:04								0:00 / 0:04	

FIGURE 4.20: Screenshot of part of the web interface used for the listening tests

As shown in the interface, for each pair of sounds, the listeners were instructed to select one button, on a scale from 0 to 3, according to their preference about the naturalness of the presented sounds. If the left sound sounded much more natural than the right one, the listener should thus select the leftmost "+3" button (or the other way around). If the left sound sounded only slightly more natural than the right one, the listener should select the left "+1" button. And so on ... The listeners were also asked to concentrate only on the perceived pitch fluctuation (and not on other aspects like the timbre). A high CMOS score (such as those shown in figure 4.21) for certain stimuli denotes a preference for those stimuli over the others having lower scores. The perceived difference between 2 types of stimuli can be considered as being really significant if the confidence intervals (the whiskers in figure 4.21) don't overlap. All pairs of sounds were presented in random order. The original test with the full instructions can still be found at url[3].

4.4.5.1 Test I: jitter and micro-prosodic components

The first part of our test was aiming at evaluating the usefulness of modeling both jitter and micro-prosodic components to improve the naturalness of the synthesized voice, and validating the approach used for generating those 2 layers. For this purpose, 2 very simple examples with long sustained notes were generated for each voice. One of the examples consisted of a non-sense sentence comprising only vowels and voiced consonants sung on a single note; the 2nd example was an actual French sentence sung on a very simple melody with no expression. The pitch was adapted to the mean frequency of each database in order to avoid important transposition ratios that may degrade the sound quality of the synthesis. For each example, a flat version (i.e. with a fixed f_0 on each sustained note) was compared to the same example with only the jitter component, only the micro-prosodic component, or both layers.

[2] http://recherche.ircam.fr/anasyn/ardaillon/ardaillon2015f0model/

[3] http://recherche.ircam.fr/anasyn/ardaillon/Test2015luc/index.php
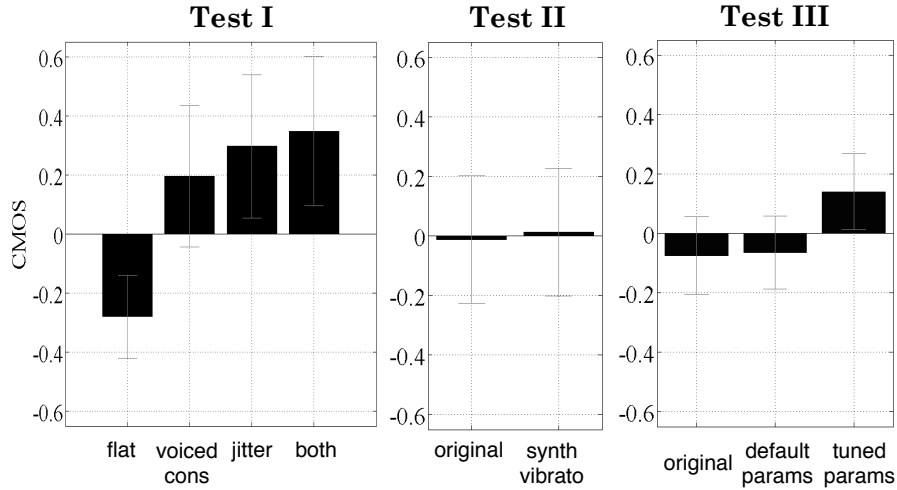
FIGURE 4.21: Results of a listening test evaluating the naturalness of the proposed f_0 model

According to the results showed in figure 4.21, we can conclude that adding either the jitter or the micro-prosodic ("voiced cons.") component to the curve clearly improves the perceived naturalness of the voice, at least in the case of sustained notes without vibrato. The jitter seems to have slightly more impact on the naturalness that the micro-prosodic component, which is understandable, as the jitter is present everywhere whereas the inflections of the micro-prosodic component are only localized at the positions of the voiced consonants. We can also observe that the addition of both layers together improves a bit further the naturalness.

4.4.5.2 Test II: vibrato model

In the second part of the test, we aimed at evaluating the relevance of our simple vibrato model, using a fixed rate f_{vib} and an ASR piece-wise-linear amplitude curve. We thus extracted the f_0 contours of 4 short songs' extracts (of 5 to 8 seconds), from various singing styles (lyrical singing, pop, and French variety), sung by the 2 same singers who recorded the databases MS and RT. We then marked the boundaries of each vibrato parts and flattened manually the f_0 curve through the vibrato cycles using Audiosculpt[4] [Bog+04]. Then, we generated new vibratos for each of the flattened notes, adjusting manually the model parameters (rate, extent, attack and release durations, and initial phase when required) trying to match the original vibrato with the model, and we added these new vibratos on the flattened curve. We then resynthesized each of the analyzed examples with both the original curve and the modified curve with synthetic vibrato. In order to apply the real f_0 curves coherently with the aligned lyrics, we also extracted the phonemes' durations from the original recordings using manual annotations, and applied them to the synthesis. The subjects were asked to focus on the vibrato sections for this test. The vibrato segments in those examples were from 0.14 (1 cycle) to 2.17 (12 cycles) seconds long.

[4] http://forumnet.ircam.fr/product/audiosculpt-en/

As can be observed on figure 4.21, the listeners showed no preference between the original vibrato and our simple model. This confirms that such a simple vibrato model seems sufficient for generating natural singing voices' f_0 curves. This is also coherent with the conclusions of [MB90], and suggests that a more precise characterization of the vibrato shape, amplitude, or frequency, is not necessary. However, it would nevertheless be easy to introduce more variations in our model, for instance by slightly randomizing the knots positions and weights values during the sustains segments for the B-splines-generated vibrato.

4.4.5.3 Test III : complete f_0 model

In the last section of our listening test, we evaluated the potential of the complete model (all 4 layers summed together, the vibrato being modeled as a separate layer for this test) for natural singing voice synthesis, by comparing synthesis using the generated f_0 curves to real ones extracted from recordings. For this purpose, we analyzed and carefully corrected the f_0 curves of 5 extracts of various singing styles sung by both our singers (among which the 4 extracts used in test II). For each extract, a score was created specifying the midi notes to be sung. Similarly to the test II, the original phonemes' durations were used for the synthesis, and the notes' onsets were thus aligned to the positions of the vowels. We then generated 2 different f_0 curves from this score, using our model: the first one using manually chosen default parameters which were the same for all transitions, attacks, releases and vibratos segments to globally approximate the original variations; the second one refining manually the tuning of the parameters, for each transition, attack, release, and vibrato independently, in order to better match the original curve locally. For each example, 3 versions were then synthesized using the 3 curves ("original", "default params", and "tuned params") and each pair of those 3 versions were compared regarding the perceived naturalness. In order to keep the test short enough and allow listeners to easily compare the sounds, only short extracts of 4 to 8 seconds were used, and only 2 randomly chosen extracts were selected for each voice and presented to the listener.

In figure 4.21, the results of this 3rd test show that the subjects were not able to make a difference between the original curve and both the generated curves (the confidence intervals are highly overlapping). Thus, the main conclusion that can be drawn is that the used model seems appropriate to generate natural f_0 contours for singing voice synthesis, for various singing styles, provided that the tuning of the parameters is appropriate. The positive tendency for the "tuned params" versions may be explained by the fact that the generated curve, driven by the midi notes in the score, may correct eventual mistuned notes in the original version. However this tendency is not very significant. The fact that no difference is made between the "original" and the "default params" versions is quite encouraging for the automatic modeling of f_0 curves with few need for manual tuning. However, one may also expect that a difference would be made for longer examples with more variations, as the default parameters wouldn't be able to reproduce the variety of expressions accross time and might start to sound too mechanical.

4.5 Intensity modeling

Compared to the modeling of the f_0, fewer works address the problem of explicitly modeling intensity variations for expressive singing voice synthesis. Opposed to f_0 fluctuations, one may assume that the intensity variations carry less expressive

features. However, singers nevertheless control the intensity of their voice to give more emphasis on specific notes or parts of a song, and shape the dynamic of a song using expressive gestures like crescendi or decrescendi at various levels (e.g. on a single sustained notes, or at the phrase level).
In HMM-based systems, the intensity might be implicitly modeled in the spectral envelope, usually reconstructed from MFCC coefficients, or explicitly based on the estimated power of the signal (similarly to the pitch), as in [STK10]. The Bézier curves in [MBM06] are also used to model the energy contours, along with the f_0 curve. As evoked in section 2.6.2, units concatenation may also be used to generate appropriate dynamic contours, similarly to f_0, as is done for instance in [UBB13a] based on smoothed energy contours analyzed from a database of recordings.
The human auditory system is less sensitive to fine energy fluctuations than to f_0 variations, and the control of intensity thus requires less precision. It is thus possible to obtain reasonable results for the control of dynamics by roughly drawing an energy contour by hand to apply some expressive gestures on sustained notes, as can be done for instance in the Vocaloid software using the provided GUI. However, it is still useful to be able to quantify those variation for easing the task of manual tuning and characterizing intensity variations according to singing style for instance. From this point of view, the above-mentioned approaches to intensity modeling suffer from the same drawbacks than when applied to the modeling of f_0, that is, mainly a lack of intuitive control parameters.

As said in section 2.6.2, the intensity variations are also closely related to timbre and phonetics, as every consonant induces micro-prosodic fluctuations of the energy. However, as our synthesis system is based on units concatenation, the diphones used already implicitly contain those variations which thus don't need to be modeled. We thus only focus here on the expressive intensity-related gestures carried by sustained vowels.
[Jen99] presented a simple parametric model of amplitude envelopes for isolated partials of musical sounds, using 5 segments (start, attack, sustain, release, and end segments). In a similar idea, we use a very simple parametric model of intensity defined at the note level, on the time-span of the vowel. For each note, a simple Attack-Sustain-Release (ASR) curve is used, as shown in figure 4.22, similarly to the amplitude envelope of the vibrato shown in figure 4.8. This envelope is bounded in time by the vowel's onset and offset in the corresponding note. 5 parameters are used for each note (each vowel), which are: a maximum value I_{max}, the attack durations d_a, the attack depth (as a ratio of the maximum value) $0 < \alpha_a < 1$, the release duration d_r, and the release depth α_r, as shown on figure 4.22. Then, on consonants, we simply interpolate the values at the end and beginning of the surrounding vowels to get a continuous curve. In case of melisma (several successive notes sung on a single vowel), the attacks and releases on each note are adjusted so that the curve is continuous: if the right note is louder than the left one, the release's depth and time of the left note are set to 0 and the attack's depth of the right note is set according to the loudness difference between the 2 notes, with a minimum duration of 0.2s; if the left note is louder than the right one, the release of the left note is adjusted similarly and the attack's depth and duration of the right note are set to 0. We found out that this simple model can reasonably fit the loudness contour of most vowels in real singing recordings. For generating the curve, default parameters can be used, that may then be manually adjusted by the user (using a dedicated xml file, similarly to the f_0 parameters). The next chapter will also address the problem of choosing appropriate intensity parameters for each note according to its

context. Note that in our system, the intensity is defined as the measured loudness (in sones) of the synthesized signal (based on a simplified loudness model that will be detailed later in section 6.2.3.1).
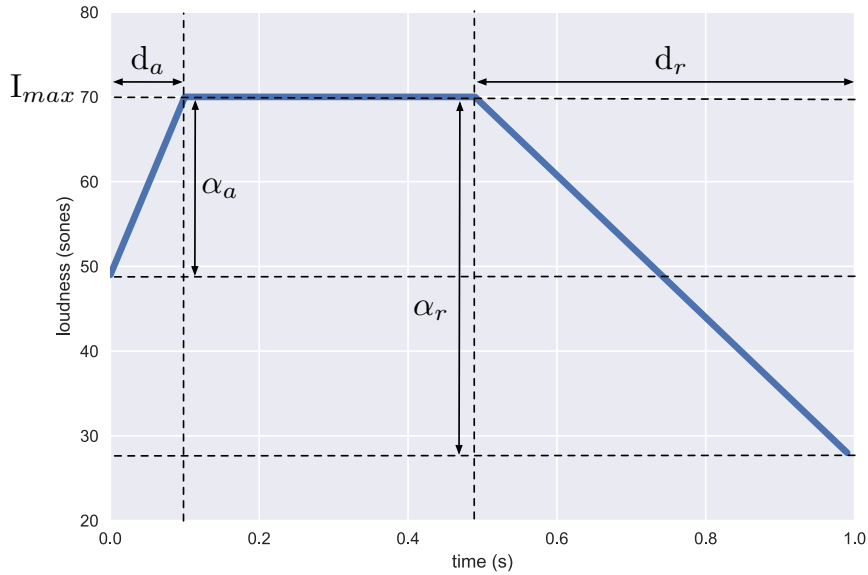
FIGURE 4.22: Parametric model of intensity profile for a vowel in a single note.

Additionally, some tremolo may also be added using a simple sinusoidal modulation (possibly synchronized with the vibrato).

4.6 Summary and conclusion

In this chapter, we detailed the approaches used for generating the synthesis control parameters in the control module. We first explained how the phonetic transcription of the lyrics can be specified. Then, from the given phonetic transcription and musical score, we detailed the rules and models used for generating the 3 main control parameters to be used as input by the synthesis module, namely the phonemes' durations (and positions), the f_0, and the intensity.

The main contribution described in this chapter is the development of a new parametric f_0 model for singing voice, offering intuitive controls for the user to shape the expression, using B-splines. The features controlled by the model are: preparations, overshoots, vibrato, transition's durations, and attacks and releases' depth and durations, thus encompassing the main expressive f_0 fluctuations used by most Western-European singers. The main advantages of this model over other state-of-the-art approaches like HMM or units concatenations are that it can be used without any data, only using default or user-defined parameters, and that the resulting curve can thus be intuitively modified by the user.
Another advantage is that some rules can be easily applied, for instance to ensure more coherence between the placement of the transitions and that of the phonemes, which appears to be very important to obtain a natural result.

Some subjective listening tests were conducted to validate this model, which showed that adding jitter and micro-prosodic fluctuations improve the perceived naturalness of the synthesized voice. Then it appeared that listeners did not show any preference between natural vibrato and a synthetic model consisting of a perfect sinusoid scaled by an ASR envelope, thus validating the use of this simple parametric model for synthesis. Finally, the full model has been confronted to real f_0 contours and the results suggest that it is appropriate to generate natural-sounding f_0 curves across a variety of musical styles. This contribution resulted in a publication to the Interspeech conference [ADR15] (although the proposed model has been further improved after this publication).

Chapter 5

Modeling singing styles: towards a more expressive synthesis

5.1 Introduction

In the previous chapter, we introduced the control module of our synthesis system, in charge of generating the phonemes durations and the f_0 and loudness profiles from the score and lyrics, and showed that the proposed f_0 model can produce natural-sounding f_0 curves for several singing styles. However, we assumed so far that all the control parameters were already known, using default settings or manual tuning, and no automatic way for choosing those parameters was given. Introducing context-dependant variations can make an interpretation more natural and musically interesting. However, even though the provided parameters are assumed to be intuitive for the user, their manual tuning remains a tedious task, that may also require some expertise to obtain a satisfying result. It is thus desirable for the system to be able to automatically choose the most appropriate parameters according to the local and global contexts defined by the score.

While some stylistic characteristics are inherent to the score itself (melody, rhythm, tempo, ...), a single score can be interpreted in many acceptable ways, and the precise time evolution of the control parameters (and especially the f_0) also carry an important part of the stylistic characteristics. In similar contexts, the most appropriate parameters are thus likely to change from one singing style to another, which should be considered when generating those parameters, in order to mimic the various strategies used by different singers.
The extraction of the control parameters from recordings of a specific singing style, along with the associated contexts, should thus allow to catch some of the expressive characteristics related to this style, while preserving some variability and the coherence necessary to produce a natural-sounding singing voice.

2 goals are therefore pursued in this chapter:

- generate a more expressive synthesis with a minimum of manual tuning, by automatically varying the control parameters according to the musical context;
- model various singing styles within a single framework, based on data extracted from recordings.

In the following, we will start by discussing the definition of a singing style and the various aspects implied in its perception. Then, based on those considerations, we present a corpus of recordings and annotations that has been constituted for the purpose of singing styles modeling. We then propose an approach to build

different style models from this corpus, that are then used to generate appropriate control parameters at the synthesis stage, according to a target score. Finally, we will present some subjective listening tests that have been conducted in order to evaluate the proposed approach, and discuss the obtained results.
This work was conducted in collaboration with the musicologist Céline Chabot-Canet (specialist of the interpretative styles of French singers from the 2nd half of the XXth century), author of [Cha13], who helped us by providing some insights into the specific features that may characterize the style of a singer, along with the contextual factors that may influence his/her interpretative choices, and in the constitution and annotation of our stylistic corpus.

5.2 Singing style: definition and implications

Defining what is a music style, and more specifically a singing style, is a difficult task, as it implies to categorize, within a set of subjectively defined classes, an infinite continuum of possible musical and vocal features relying on high-level and low-level objective and measurable characteristics (such as melody, rhythm, ornamentations, f_0 variations, timbre, ...), but also involving more subjective features such as the target audience (e.g. children songs), the emotion conveyed (e.g: sad or joyful), etc... Authors of past studies have been defining singing styles in various ways, considering different aspects to form categories like: "opera", "children's songs", "angry/sad/happy", or "jazz standards" (as listed in [Umb+15], table 4), thus referring to either a well-known broad musical category, the age of the audience, or various emotions. Singing style may thus be simply understood as: "what makes one performance different from another one, and what makes two different performances sound similar". But such definition doesn't allow to characterize a specific style.

We all know many broad stylistic categories, such as "rock", "pop", "jazz", "soul", "metal", "classical music", and many more... However, although the voice may carry some important characteristics that make those genres recognizable, these terms are not specific to singing and are based on many aspects that go well beyond the properties of the voice itself, relating to the various musical instruments or sounds used. Furthermore, these categories don't have precisely defined boundaries. On one hand, some songs may be considered to be a mix of several of those categories, but they may also be further classified in sub-categories of these musical genres. For instance, the "rock" category could be subdivided into "garage", "progressive rock", "hard rock", ... Similarly, the term "jazz" may encompass "swing", "New Orleans", "be-bop", etc... And even when considering only the vocal part in one of those sub-genres, each singer has its own typical voice timbre and expression that make him well identifiable among others. But conversely, some singers may also sing rather differently from one song to another. According to these considerations, it appears that a singing style can be defined at many levels, and it is thus difficult to come up with a unique and universal definition of what is a singing style.

As we are planning here to model singing styles for synthesis purposes, based on data from real recordings, the question of precisely defining what is meant by the term "singing style" is nevertheless important. Whether the system is

based on some machine learning approach, units concatenation, or user-defined rules inferred from the observation of the recordings, the gathered data should be consistent with this definition and present sufficient regularity in the features to be modeled.
First, we will start here by discussing the various aspects and features that may be involved in the perception of a singing style. Then we will propose a more specific definition for the purpose of singing synthesis and discuss which features can be modeled in our framework.

5.2.1 Singing styles: perceptual models and aspects involved

Many aspects are involved in the perception of the singing style from a recording. All those aspects have been thoroughly studied in [Cha13], focusing on the analysis of the interpretative styles of 59 famous French singers of the XXth century. In [Cha13] and [Cha08], the author identifies 3 "models" that contribute, from the perception point of view, to the internal representation of a singing performance for a listener:

- **The "score model":** related to the properties that allow us to identify the song being interpreted, which are mainly the melody, the rhythm, and the lyrics.
- **The "generic model":** related to the identification of a musical style into which the singer or the song would be categorized. This model relates to the musical arrangement and the instruments used, the treatment of the voice on the recording (audio effects, ...), the use of some recurrent vocal effects (growl, ...), some degree of liberty between the notes and rhythm written in the score and its interpretation (e.g: agogic rhythmical deviations, addition of ornamental notes, ...), and some elements of phrasing (use of legato, crescendi, ...).
- **The "stylistic model":** related to the identification of a specific singer. It thus implies the specific timbral characteristics of the singer's voice and some recurrent interpretative vocal effects used by the singer, but also, similarly to the generic model, some phrasing elements and degree of freedom relatively to the interpreted score, and some treatments applied to the voice, along with the instrumental accompaniment used.

We summarize below the various aspects upon which those models rely, which can be divided into 6 categories:

- **Instrumental aspect:** A song is rarely sung a cappella and the voice is usually accompanied by instruments which thus have a great impact on the perception of the interpretation and the music style, by embedding the vocal part in a larger context. The instrumental aspect encompasses both the instruments used and the musical arrangements (the chords, the times where each instrument plays or not, counter melodies, ...). One may thus wonder if a music style is perceived to a similar degree when listening only to the voice or to the whole musical content.
- **Technological aspect:** We denote here by the term "technological aspect" all characteristics related to the technological means used for recording and

processing the voice signal. This encompasses the whole sound chain, from the microphones used and their placement in a specific room (which may have natural reverberation), to the techniques used to write/read the recorded sound on/from a physical support (mechanical or digital). The first microphones used for early music recordings and the machines used to write music on vinyl discs or cylinders and read it, such as the phonograph or the gramophone, give to the music of this period a specific well-recognizable sound that contributes to the perception of the music style.
Additionally, many analogical or digital effects, such as reverberation, delay, compression, or equalization, can be used to process the voice, either to better integrate it into a mix, or to give it a specific timbre that is characteristic of the musical style.

- **Linguistic aspect:** Several things can be considered regarding the linguistic aspect of a song. First, the text itself may be used expressively, for instance using alliterations to emphasize the rhythm by repeating similar timbral characteristics on each notes, which is a typical effect used in rap music for instance.
  Then, the way of pronouncing certain phonemes may be important to consider for some singers. For instance, the way the famous singers Georges Brassens, Edith Piaf and Jacques Brel pronounce the phoneme /R/ is a salient characteristic of their respective personal styles.
  Finally, certain music styles may be closely related to the use of a specific language. It is for instance not likely to find recordings of Beijing opera recorded in another language than Chinese. Similarly, gospels are more likely to be sung in English rather than French.
  The meaning of the lyrics may also be an important aspect for some music styles.

- **Symbolic aspect:** The symbolic aspects relates to all the informations that have been defined by the composer of the song, mainly including the lyrics, the melody (notes' pitches) and rhythm (notes' durations).
  The use of lyrics structures the song at several levels. For instance, the division of the text into verses and chorus, typical in some music styles, imposes a certain macro-structure to the song, while smaller levels like the syllable impose a micro-structure strongly related to the rhythm of the song.
  The tempo, rhythmical patterns, melody, musical scale, chords progressions, or pitch range are various examples of features embedded in the score that are strongly related to specific music styles, independently of the interpretation of a specific singer. For instance, the very high pitch range used by lyrical soprano singers is not well adapted for other styles like rock; blues uses some specific sequences of chords that are not really transposable to another music style; etc ...
  An important consequence is thus that scores are often not well adapted for being interpreted in different singing styles, which should be taken into account for evaluation purposes, as will be further discussed in section 5.7.
  Singers also often don't strictly follow the notes in the score. They may for instance add some ornamental notes, or deviate from the exact durations of the score, by lengthening or shortening some specific notes, or make use of anticipations and ritards relatively to the theoretical notes' onsets, or systematic deviations like swing, and continuous tempo variations (accelerando or

rubato). Sustained notes may also be shortened compared to their theoretical duration.

- **Prosodic aspect:** Although the term "prosody" is usually rather used for describing speaking voice, we denote here by this term all the variations that apply to the 3 basic acoustic dimensions that are phonemes' durations, f_0, and voice intensity, which have been already studied in the previous chapter. The evolution in time in these 3 parameters highly relates to the specific phrasing and interpretative effects of a singer. Contrary to all the previously mentioned aspects, this prosodic aspect is also closely related to the level of proficiency of the singer.
  As already said, the f_0 is especially important, as it carries the melody and acoustic features specific to singing voice, such as vibrato, preparations and overshoots. In [NLM07], [Kak+09], and [Pan+17], the authors proposed some approaches to singer and singing style identification based on the f_0 contours, obtaining good accuracy, showing that the f_0 (and especially the vibrato) carries particularly important characteristics related to singing style and singer's identity, based on local dynamic variations independently of the musical information from the score. [SG09] stated that the perception of the quality of a voice is more influenced by changes in the f_0 after vocal training rather than the spectral characteristics, vibrato having the most important contribution. Other features related to the f_0 are the use of portamento, or glissando, and the pitch accuracy.
  Some variations related to the intensity, or dynamic, are crescendo and decrescendo, dynamic contrasts (from one note to another), or tremolo. In [Kin+14], a visualization of singing styles in a 2D space based on pitch and dynamics trajectories is proposed to identify some characteristic vocal gestures of the singers from those trajectories.
  In [NT10], the authors showed that infant-directed performances have fewer expressive variations in timing but greater dynamic modulations than non-infant-directed performances.

- **Timbral aspect:** Compared to many instruments, the voice has a very flexible timbre that have many implications on its perception. Each singer has its own pitch range and formants' structure, related to its physiology, which partly confers its identity to the voice. The source characteristics also provide an important contribution to the singer's voice identity, which can be for instance more or less breathy or tense. But beyond the singer's identity, some voices are better suited for certain singing styles than others. This should be considered for synthesis purpose, as the style may not be similarly perceived depending on the voice used for producing the synthesis.
  Besides features intrinsically related to the singer's physiology, the timbre may also be expressively modified, either permanently or punctually by the addition of specific effects, depending on the singing style.
  Several works have been dedicated to the studies of the specific timbral features of various singing styles. For instance, the well-known singer's formant, characterized by a prominent peak of the spectral envelope around 3kHz obtained by modifying the larynx position, is a typical feature of classical operatic singing [Sun01; Nak04]. Another timbral feature is the vocal register (or laryngeal mechanism: fry, chest, head, falsetto) that may be used either permanently or punctually to produce specific effects (e.g. in yodel [Wis07]). A singing technique specific to some non-classical styles

of singing, such as pop or musicals, is called "belting", characterized by a loud and bright voice with a consistent use of the chest register and a reinforcement of the 2nd harmonic by the 1st formant [SM93; BS00; TW09]. [BK06] highlights some differences related to resonances, loudness and spectral slope between 3 types of Croatian folk singing. In [TS01], the differences in the source's characteristics of a single female singer in classical singing, pop, jazz and blues have been compared, demonstrating the possible adaptation of the voice timbre according to singing style, despite the singer's physiological constraints. Similarly, the differences in vocal source and resonances between soul and musical theater have been studied in [HLS17] based on acoustic measures and spectrographic analysis.
Finally, certain vocal qualities are typically used in certain music styles as punctual expressive effects. For instance, the growl effect and other types of rough phonations are used a lot in styles like blues or rock, among others (e.g. by singers like Ray Charles or Louis Armstrong [Pfl10; Sak+04]).

Note that some expressive features may also be considered as transverse, combining several aspects simultaneously, as suggested in [Umb+15]. An example are the attacks of notes, that may combine a rise of f_0 and intensity as well as the use of specific timbral characteristics (e.g. fry, breathy, with a glottal impulse, ...).

Based on those descriptions, one can state that: the score model is mainly related to the symbolic and linguistic aspects; the generic model to the instrumental, technological, timbral, and (to some extent) prosodic aspects; and the stylistic model to the technological, instrumental, timbral, and prosodic aspects.

5.2.2 Singing styles for synthesis: definition and modeled features

Now that we have reviewed the many features that may be involved in the perception of music and singing styles, we may discuss to which extent those elements can be modeled in the framework of our singing voice synthesis system. As we have seen, the aspects involved are very diverse and many are not directly related to the voice. One may thus only expect to model parts of those aspects with our system. We detail below which aspect have been considered in our work, and then propose a more precise definition of what should be understood by the terms "singing style" in the rest of this chapter.

The instrumental aspect will obviously not be considered in this work, as it doesn't imply the voice.

The technological aspect is also not dealt with, besides the recording of the databases (which may impact the sound of the voice for concatenative synthesis, depending on the recording conditions that have been previously described for our databases in section 3.2.1.4). The addition of some audio effects after the synthesis is left to the choice of the composer or sound engineer. Besides the stylistic considerations, note also that the simple addition of a subtle reverberation, by simulating a natural environment, can have a significant impact on the perceived naturalness of the voice, compared to a raw synthesis.

Regarding the linguistic aspect, we already discussed how the lyrics may be input and possibly automatically phonetized in the previous chapter. We assume that the lyrics are fully determined by the composer. However, as we have seen,

the use of a specific language or different pronunciations of a same phoneme (e.g. /R/) may have to be considered in relation to the singing style. For this purpose, a mechanism has been implemented in our system in order to deal with some extensions of the databases to allow to add some new pronunciation variants for a specific phoneme, or add missing phonemes for a specific language. Specific tags in the input lyrics can then be used to choose the proper pronunciation. We used for instance this system to produce some synthesis in English, based on our French databases, using specific additional recordings for english phonemes. Some tests have also been conducted for synthesizing Japanese using a dedicated restricted database.

Regarding the symbolic aspect, we saw that the score already contains many important informations related to singing styles. But we saw also that some deviations from this score should be considered, like for instance the insertion of ornamental notes and rhythmical deviations at the phrase level (rubato, accelerando, ritardando) or at the note level (ritards and anticipations). Such variations may be considered by the system and applied in the symbolic domain to modify the score, depending on singing style, before running the synthesis, for instance using rule-based approaches [FBS09]. An approach to apply expressive tempo variations in monophonic instrument phrases has also been proposed in [Gom+03], which may possibly also be applied to singing. However, this aspect has not been considered in the present work, and our system strictly respects the informations given by the input score, assuming it already contains all ornamental notes and rhythmical deviations, that should thus be precisely defined by the composer to fit the target style.

The prosodic aspect is the one we will focus on in this chapter, assuming, as has been stated in previously mentioned studies, that it carries many important stylistic characteristics. Based on the parametric models presented in the previous chapter, we propose an approach to learn the typical behaviour of singers according to the musical contexts defined by the score, for different singing styles, and apply it during the synthesis. We thus aim at automatically generating the f_0, loudness curve, and phonemes' durations with respect to a target singing style.

Finally, the timbral aspect may be considered to some extent. In concatenative synthesis systems such as ours, the specific timbral characteristics of the singers are implicitly contained in the database used. An appropriate database should thus be chosen according to the target singing style. In the databases we recorded, described in table 3.2, we have for instance the EL database that corresponds to a lyrical singing style, while the timbres of RT and MS databases rather correspond to "pop" styles. In preliminary tests, we also had recorded a restricted database with the singer RT in lyrical style (RT_lyr), and results of synthesis using this RT_lyr database preserved well the lyrical timbre quality from the database, as illustrated in sound 5.2 compared to sound 5.1 using a non-lyrical database.
Alternatively, some transformations (e.g. for age or gender[1]) or voice conversion techniques [Hub15; Lee+14] may also be applied to extend the potential timbre space covered by a single database and change singer's identity, although further researches are still necessary before obtaining satisfying results. Some ongoing research in the analysis/synthesis team at IRCAM are investigating the possibilities

[1] `http://forumnet.ircam.fr/product/ircam-tools-flux-trax-en/`

to apply voice conversion techniques issued from previous research projects [Hub15] to the output of our synthesizer to make it sound like a target singer.
Additionally, some transformations may be applied to modify specific timbral characteristics such as the tenseness, breathiness, or roughness of the voice, as previously reviewed in section 2.5. Some work on vocal roughness transformations will be presented in the next chapter. However, the automatic control of such effects is not addressed and has not been much studied yet.

As previously stated, those various features may have implications at several perceptual levels, i.e. to recognize a song, a generic style, or a specific singer. In this chapter, we focus on the modeling of the prosodic features, which have implications on both the previously defined generic and stylistic models, which means that some similar expressive prosodic features are likely to be shared among various singers of a similar generic style (e.g., jazz, rock, opera, ...). For instance, all lyrical singers use vibrato rather extensively, while rock singers don't use it much. However, in the same generic models, some of those features are also likely to vary between singers.
In this work, we aim at modeling expressive features based on a set of recordings representative of a particular singing style. For this purpose, it is thus important to have as few disparity as possible in those recordings. Learning a style using recordings from different singers may thus be a problem, as there may be too much differences in the features from the different singers, even though they belong to the same generic category, which may either lead to oversmoothing or to inconsistent expressions in the synthesis.
To avoid this problem, a singing style will thus be defined in this chapter as that corresponding to a small set of consistent recordings of a single singer, with a rather uniform and stereotypic interpretative style, chosen to be representative of a certain generic category.

In the next section, we will first describe the corpus of recordings that we used for our work, before explaining our approach to singing styles modeling.

5.3 Styles corpus

5.3.1 Choice of singers and recordings

In order to study and model different singing styles, a corpus has first been created. A possible approach for this purpose is to use dedicated a capella recordings. In [UBB13b], the author proposed an approach to systematic database creation, generating automatically a set of short melodic exercises that cover various combinations of features for a certain singing style. In [STK10], 5 japanese children songs recorded by a male professional singer in a "deep bendy" style are used. The study in [UBB13a] used 4 recordings of a female trained singer in soul and pop style, for a total of 6 minutes. In [Umb15], 17 recordings of jazz standards where also recorded for a total of around 18 minutes.
However, a particular drawback in [UBB13a] and [Umb15] is that the recordings were sung using only vowels. While this avoids the influence of the micro-prosody on the expressive features, which should better be considered separately, as is done in our multi-layer f_0 model, this also does not allow to learn the possible expressive use of phonemes' durations, which may be of importance.

A particularly interesting goal to achieve would also be to be able to model a singing style from existing music recordings, allowing for example the modeling of famous historic (deceased) singers, which is not intended in the above-mentioned studies using dedicated a capella recordings. In this direction, commercial recordings with instrumental accompaniments were used in [IIO14a] and [IIO14b] to extract vocal expressions of famous Japanese singers and apply them to synthesized voices.
In the present work, we thus also propose to study and model singing styles of specific singers based on commercial recordings. For this purpose, a corpus has been constituted, with the help of our musicologist collaborator Céline Chabot-Canet. In order to benefit from her specific musicological knowledge as well as some well-known French cultural references, we chose to base our work on recordings of 4 famous French singers from the 2nd half of the XXth century, representative of different styles, rather than using dedicated recordings from unknown singers. Another reason for this choice is that our system, in the framework of the ChaNTeR project, is primarily targeting the French language. The 4 singers we chose are: Edith Piaf, Sacha Distel, Juliette Greco, and François Le Roux.

As has been previously evoked, many scores are not well suited to be sung in different styles. In order to compare and evaluate the modeled singing styles, it is thus beneficial to find a score that may be reasonably interpreted in the styles of the different chosen singers. The choice of those 4 singers has thus also been encouraged by the fact that they all recorded an interpretation of the same song "Les feuilles mortes" ("Autumn leaves" in English) in their own singing style, thus providing a common reference for comparison. A musicological study of various covers of this song by many singers has been conducted in [Cha08], stating that it is especially suitable to be interpreted in various styles.

Using commercial recordings is more challenging than using a cappella singing, as the presence of other instruments makes the analysis and annotation process more difficult and fastidious, but it has also several advantages regarding the evaluation process. One reason is that finding a capella recordings of a same song by several singers in different styles is not easy, and conversely, it is hard for a singer to sing in different styles ("generic models") while getting rid of his/her own personal interpretative style ("stylistic model"). For these reasons, previous studies have limited the evaluation of their methods to a single style, which doesn't account for the ability of those methods to properly differentiate several styles. For evaluation purposes, one may also assume that it is preferable to model the styles of well-known singers who listeners may already be familiar with.

In the ideal case, one should be able to learn a style even from a single song, as a same singer may sing a bit differently from one song to another. However, this may lead to overfitting, and for a good generalization, the database should also ideally cover the complete space of musical contexts, in terms of possible notes' durations, pitches, and intervals. A compromise is thus to be found, using a reasonable amount of data covering various possible contexts, while keeping enough consistency between the recordings. For each of the 4 singers, 2 others songs than "Les feuilles mortes" have thus been selected. Table 5.1 summarizes the content of our corpus. The given tempi are approximative, based on manual annotations of beats (the perception of tempo being subjective, some possible

TABLE 5.1: Description of our singing styles corpus

Singer	Sex	Style	Song	Tempo (BPM)	Duration	Total duration
Edith Piaf	female	Chanson réaliste	Les feuilles mortes	70	3'29"	9'55"
			La foule	62 (or 186)	2'58"	
			Hymne à l'amour	65	3'28"	
Juliette Greco	female	Chanson rive gauche	Les feuilles mortes	64	2'58"	9'01"
			La javanaise	44 (or 132)	2'28"	
			Je hais les dimanches	64	3'35"	
François Le Roux	male	Mélodie française	Les feuilles mortes	60	4'37"	8'34"
			Dernier voeu	86	1'42"	
			Sous l'épais sycomore	85	2'15	
Sacha Distel	male	Chanson de charme	Les feuilles mortes	70	3'31"	9'13"
			Parlez-moi d'amour	108 (or 36)	2'28	
			Que reste-t-il de nos amours	55 (or 110)	3'14"	

alternative values are given between parenthesis). (Note that the given durations represent the total lengths of the songs, including sections without singing voice.)

In terms of size per singer, this corpus is comparable to those used in [UBB13a] and [STK10], whose sizes are respectively around 6' and 5' of a capella recordings. Some links to all songs of the corpus can be found on the web page at url[2].

5.3.2 Styles description

As the singing styles we propose to model here may not be well identified by everyone (and especially for non-French readers), we shortly describe here the principal characteristics relating to the voice for each one, based on the work of Céline Chabot-Canet [Cha13] (except for the style "mélodie française" of François Le Roux, which is not studied in this work, but which is documented elsewhere).

- **Chanson réaliste - Edith Piaf:**
  Concerning the timbral aspect, a particular characteristic of this style is the use of belting, with a tense and loud voice and a rather uniform timbre and energy. The overall voice timbre also sounds quite low.
  The pitch and loudness tend to be quite correlated: the higher the pitch the more tense and loud the voice is. Ascending glissendi are also often accompanied with a crescendo and descending glissendi with a decrescendo.
  The first notes of sentences are usually attacked with an ascending glissendo, and a rather continuous movement of the f_0 can be observed on groups of succeeding short notes. The highest notes of musical phrases are usually also attacked with a rather long glissando (long transition).
  In this style, the vibrato is not permanently present, but is rather used intentionally on sustained notes. An alternation between short notes without vibrato and long notes with vibrato can thus be observed. When present, the vibrato is usually wide, and very regular, especially at the end of sentences. The vibrato of Edith Piaf is particularly identifiable for being quite strong, with a high frequency (between 7 and 8 Hz).
  The text is well articulated for a good intelligibility. The phonemes /R/ are systematically rolled and often quite long. Semi-vowels are also usually long and well articulated.
  Some rhythmical deviations like ritards and rubato are regularly observed.

- **Chanteur de charme - Sacha Distel:**
  The French terms "chanteur de charme" corresponds to what would be called

[2] http://recherche.ircam.fr/anasyn/ardaillon/these/these.php

a "crooner" in English.
In this style, the voice is soft (not very loud) and languorous with a "jazzy" coloration, and breathiness is a recurrent vocal quality used to confer a seductive and sensual dimension to the voice. The voice can be rather deep and fry might also be used punctually (although not much for Sacha Distel).
The tempo is usually particularly slow.
Notes are often attacked with a portamento, using rather long attacks and transitions, starting lower than the target pitch, and notes are sometimes linked together in a continuous glissando. Groups of syllables are tied together in a same dynamic move with crescendo or decrescendo, without dynamical contrast between adjacent notes.
Vibrato is present on sustained notes, but is not very intense and might be sometimes slightly irregular. The expression used by this kind of singers can also get closer to speaking voice at certain moments.

- **Chanson rive gauche - Juliette Greco:**
  This style, also called "chanson littéraire" ("literary song" in english), is characterized by a predominance of the text, for which a good understanding of the lyrics is especially important. A particular emphasis is thus placed on the syllabic articulation, using regular and well marked attacks with an accentuation to give an impulsive and dynamic aspect in the interpretation. Such accents are also usually accompanied with an ascending glissando starting at a rather low pitch (big downward preparation or overshoot). The syllables on offbeats are often more accentuated.
  To avoid monotony related to a too high regularity, some rhythmical deviations are used (ritards, anticipations, ...), introducing a higher rhythmical complexity.
  The expression is also often rather close to the prosody of spoken voice, reinforcing the natural accents of speech, associated with a dramatization of the text and the search for contrasts. For this purpose, singers of this style also tend to punctually use a whispered or breathy voice.
  The pitch range of Juliette Greco is rather low and notes are often attacked from a lower pitch, with a short glissando. The pitch also tends to go down at the end of notes.

- **Mélodie française - François Le Roux:**
  While the previous styles can be considered as sub-genres of the very wide category called "chanson française" (French variety), the style "mélodie française", dating from the XIXth century, rather belongs to classical music and is quite close to the German Lied. It corresponds to a particular musical form sung with a text usually borrowed from a poetic work and accompanied with a piano. Famous composers for this style are Debussy, Fauré, Duparc and Chausson.
  Contrary to the Bel Canto (operatic lyrical singing), which is rather focused on the virtuosity and the search of an homogeneous purely harmonic timbre, to the detriment of the text, more emphasis is placed on the intelligibility of the lyrics in the mélodie française, favouring the clarity of the pronunciation. The voice is also less loud (more intimate) than in operatic singing, privileging the smoothness of the pitch and intensity variations, without much abrupt contrasts and with a fine control of nuances. The phrasing is thus rather legato, with crescendi and decrescendi encompassing several notes over a whole musical phrase.

Vibrato is also well present on sustained notes, and rather wide, accompanying the changes in dynamic (particularly wide on loudest notes).

5.3.3 Analysis and annotations[3]

In order to learn the various prosodic features based on the recordings, each song of the corpus has been analyzed and annotated with its phonetic transcription and the f_0 and loudness curves. In order to ease the annotation process, we tried to choose recordings with a rather light instrumental accompaniment when possible. We describe below how those analysis have been obtained.

Phonetic transcription: For the phonetic transcription, a first automatic pass has been run using the ircamAlign software [Lan+08] to create and approximately position phonetic markers with the labels of the phonemes in SAMPA notation, based on the raw text of the songs written in French. Then, this first transcription has been manually corrected using the audiosculpt software[4] [Bog+04].

f_0 analysis: For the f_0 analysis, we used for a first automatic pass the same previously-mentioned algorithm from superVP that we used for our synthesis databases. Using appropriate settings (especially regarding the f_0 range) on small sound segments, this already allows to have reasonable results when the instrumental accompaniment is not too much present. Then, this curve was manually corrected in audiosculpt [Bog+04], by drawing over the fundamental on the spectrogram. Depending on the instrumental accompaniment, some higher harmonics of the voice may be clearly visible, while the fundamental is hidden by the presence of other instruments. A functionality of audiosculpt allows to multiply the f_0 curve by a certain factor, which we can use to draw over the most visible harmonics before transposing the curve back to its original range. Using this approach, we could obtain a rather reliable analysis for the f_0.

Loudness analysis: In order to estimate the loudness variations of the voice, it is necessary to get rid as much as possible of the influence of the other instruments. For this purpose, it is possible, in audiosculpt to run an harmonic partials analysis, based on the f_0 curve, and then resynthesize the sound using only the partials from this analysis, which should mainly contain the voice part. The principles and algorithm behind this harmonic partials analysis have been described in [Bon+11]. Some refinements for a better estimation of the partials amplitude for non-stationary (frequency-modulated) signals have also been proposed in [Röb08], which are included in the implementation used by audiosculpt. An example of such resynthesis from the harmonic analysis is given in sound 5.3. Then, a loudness analysis can be run on this approximate resynthesis of the vocal part. The loudness model used for this analysis will be detailed in the next chapter, section 6.2.3.1. However, due to the overlap between the voice and the instruments for some of the partials and to small f_0 analysis errors, part of the voice may be missing, and part of the instrumental content may still be present, which affects the result. The obtained loudness curves have thus then been manually corrected when necessary, based on the subjective perception of the loudness variations.

[3]The annotation of the corpus has been done in collaboration with Céline Chabot-Canet
[4]http://forumnet.ircam.fr/product/audiosculpt-en/
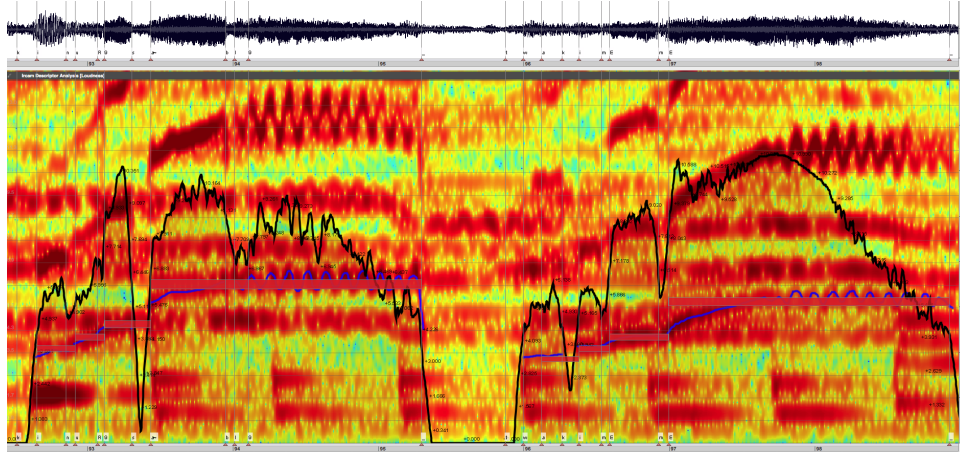
FIGURE 5.1: Example of annotation for an extract of "Les feuilles mortes" by Edith Piaf, showing f_0 (blue curve), loudness (black curve), phonemes segmentation (vertical lines), and midi notes (red horizontal bars)

Besides those 3 parameters, additional annotations relating to the symbolic musical informations from the song are necessary to derive various contextual factors that may influence the interpretative choices of the singers, and that will thus be used when learning the singing styles from this corpus, as will be explained in the next section.
Assuming that the notes' onsets coincide with the vowels' onsets, the rhythmical information (notes' durations) was obtained directly from the phonetic transcription, by keeping only the phonetic markers labelled as vowels or silences. The approximate tempo of each song was also obtained, based on the median inter-onset time of manually annotated beats. The annotation of the notes' pitches (in midi value) has then been performed manually to avoid errors. Note that in other works like [STK10], manual annotations of the f_0, phonemes, and midi events were also used. Note also that for the interpretation of "Les feuilles mortes" by Edith Piaf, only the refrain, which is sung in French, has been fully annotated, the rest of the song being sung in English. All other songs have been fully annotated.

A visualization of the various analysis and annotations in audiosculpt is shown in figure 5.1, for an extract of "Les feuilles mortes" by Edith Piaf.

5.4 Proposed approach

5.4.1 Overview

In [Shi+01], the authors suggest that *"the concept of a style implies a set of consistent features"*, that *"lend themselves to quantitative studies and modeling"*, and that *"personal style is conveyed by repeated patterns of [the prosodic] features occurring at characteristic locations"*. Modeling a singing style would thus imply being able to appropriately capture these features along with the characteristic contexts where they occur.
We already reviewed in section 2.6 the various existing approaches to expression control and singing style modeling, that all provide different means to generate control parameters according to the contexts defined by a target score. Each of the methods presented so far however has its own advantages and drawbacks. A

particular drawback of rule-based and units selection approaches is their lack of flexibility, as only a restricted and fixed set of contextual informations is used in empirically-defined rules and costs functions, which doesn't allow to easily include new contexts or to take into account the possible variable importance of some contextual factors from one style to another. However, units selection has the advantage of reproducing accurately the expressive variations of singers. Rule-based approaches also have the advantage of introducing expert musical knowledge into the system, but require a thorough musicological study to define new rules for each singing style. Conversely, HMM-based approaches relying on automatic context clustering offer more flexibility, with the possibility to use many contextual factors, and can easily model a new style using an appropriate database without requiring specific knowledge, while allowing a global high-level control of the expressivity. But they may suffer from oversmoothing and don't allow to reproduce the fine details on the f_0 or intensity contours like with units selection. Another limitation in most approaches is also a lack of intuitive and local control over the expressivity for the user.
In order to overcome the limitations of exiting approaches, it would thus be interesting to combine different methods while keeping only the best of each one. In this direction, the hybrid approach proposed in [Umb15] tries to combine some advantages of the HMM-based and units selection-based approaches. However, this approach still doesn't provide any control to the user to modify the expressivity. The use of parametric templates, as is done in [IIO14b] may however overcome this problem.

Building up on this idea, we thus propose here to combine the use of a rich set of contextual factors (with automatic decision tree-based context-clustering as in HMM-based methods) with the use of our parametric f_0 and loudness models to approximate real contours, while providing some local control of the expression and the possibility to use some rules to constrain the result based on specific knowledge.
For this purpose, the f_0 and loudness-related parameters and phonemes' durations are first extracted from our corpus, and stored in styles databases as parametrized templates, along with their original contexts. In a first learning stage, decision trees are then built from each style database to automatically cluster similar f_0 model parameters, intensity model parameters, or phonemes' durations according to their original context. The databases constituted for each singing style along with the trees built from those databases form our models of the singing styles.
At the synthesis stage, those styles models can then be used to choose in the database the most appropriate parameters according to the target contexts from the score, based on the decision trees. For this purpose, the appropriate tree is browsed from its root to the leaf corresponding to the target context, and a parametric template (or a phoneme's duration) is picked from all the occurrences associated to this leaf, and used for the synthesis. The selection of a specific template (e.g. an f_0 transition or sustain, or the loudness profile of a note), keeping all its parameters tied together, rather than a statistical modeling as is done in HMM-based systems, avoids a possible oversmoothing of the parameters. This approach is illustrated in figure 5.2 for the choice of a transition's parameters.

In our first implementation of this approach, a random selection was used in order to choose a template among the different occurrences on the leaf of the tree corresponding to the target context, as explained in [ACR16b]. However, this may
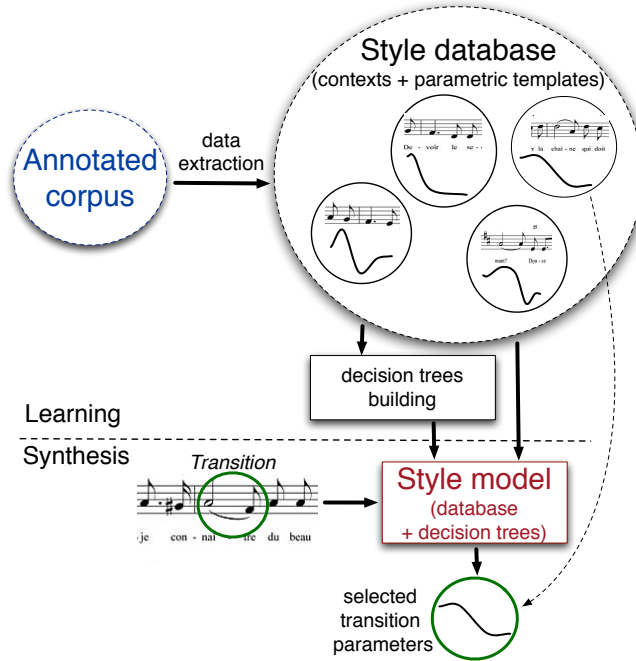
FIGURE 5.2: Overview of the proposed approach (Example for the choice of a parametric f_0 transition template)

not always lead to the best choice, as the leaf may sometimes still contain disparate values. In order to further improve the selection process, we thus compute a simple distance between the original and target contexts for all the templates of the leaf, and choose the one with the smallest distance. We used as a distance measure the sum of the absolute difference on all contextual factors normalized by their maximum absolute values (all contextual factors being thus considered of equal importance in this case).

For the proposed approach to work, mainly 2 conditions have to be fulfilled: the use of an appropriate parametric representation for the f_0 and loudness to match the real contours from the recordings; the use of appropriate contextual factors, with a sufficient coverage in the corpus.
The first condition has been already addressed in the previous chapter. In the following, we first explain the mechanism used to build the decision trees, and then discuss the various contextual factors that may be considered for this purpose. The estimation on the various parameters on the recordings, and the specificities related to the modeling of each parameter will be addressed in the next sections.

5.4.2 Decision tree-based context clustering

Ideally, if our corpus would cover every single possible combination of contextual factors for each style, one could simply choose the parameters associated to contextual factors that perfectly match the target contexts of the score for synthesis, as suggested in [Boh+91]. But having such an extensive coverage of the contexts is absolutely unrealistic, especially regarding the small size of our corpus. As previously evoked, a solution employed in some systems like [UBB13a] is to define a cost function to compute a distance between the target contexts and the original contexts from the corpus and choose the parameters associated with the closest contexts. But this approach enables to use only a

restricted set of contextual factors, and requires to make some hypothesis on their relative importance, independently of the singing style to be modeled. An alternative approach to circumvent these limitations is to use decision trees to automatically cluster together groups of contexts for which singers use similar expressive features, selecting first the most influential contextual factors among all the possibilities.
This approach can't be applied in the case of a unit selection-based system, as the contours are not quantified, but is well-suited in our case as we rely on parametric models for the f_0 and loudness.

Decision tree-based context clustering is a supervised machine learning method that may be used either for solving classification or non-linear regression problems. In our case, the problem to be solved is a regression problem. The goal is to create a model that predicts the numerical values of the parameters by learning simple decision rules inferred from the data (as a set of "if-then-else" conditions giving rise to binary choices) based on the values of input contextual factors.
To construct a binary decision tree, we thus begin with a collection of data, which in our case consists of either the phonemes' durations, the f_0 model's parameters, or the intensity model's parameters extracted from our corpus, along with a set of associated contextual factors, for a specific singing style. This collection of data constitutes the root node of the tree, where all possible contexts are represented. The tree is then built in a top-down iterative fashion, by sequentially splitting the data at each node into two new smaller subsets, on the basis of binary questions about the context. The question at each node of the tree is chosen automatically so that the homogeneity of the parameter to be predicted (e.g. the consonant duration or the vibrato amplitude) on both child nodes is maximized. A typical measure of the homogeneity used for this purpose is the Mean Squared Error (MSE). Several algorithms exist to automatically build such trees [Ode95; Boh+91; Qui93; Bre+84]. In this work we used an optimized implementation of the CART algorithm [Bre+84] from the sklearn python package[5]. More detailed explanations on this algorithm are given in appendix B.
An example of decision tree is illustrated in figure 5.3 for a simple hypothetical case where the vibrato amplitude is predicted based on a few musical contexts.

In order to avoid overfitting, a stopping criteria must be provided to the algorithm to stop the tree growth at some point. A possible criteria is to terminate splitting when the number of samples at a node falls below a given threshold, or when a maximal allowable depth is reached. According to those criteria, the final tree is obtained when none of the terminal nodes can be further split.

In case several correlated parameters are to be predicted, it is possible to use "multi-variate", or "multi-target" regression trees [Bor+15], in which case several parameters (e.g. all transitions' parameters A_L, d_L, A_R, and d_R) are clustered together simultaneously in a single tree. In such case, the only difference in the algorithm to build the tree is that the variance on the child nodes must be minimized for all parameters simultaneously (e.g. by computing the sum of the MSE over each parameter to find the best split). In our approach, this allows to keep the parameters of each f_0 or loudness segment tied together as parametrized templates. However, as the parameters to be predicted may be of different natures, and thus have values of different orders (e.g: Hz, cents, seconds,

[5] http://scikit-learn.org/stable/modules/tree.html
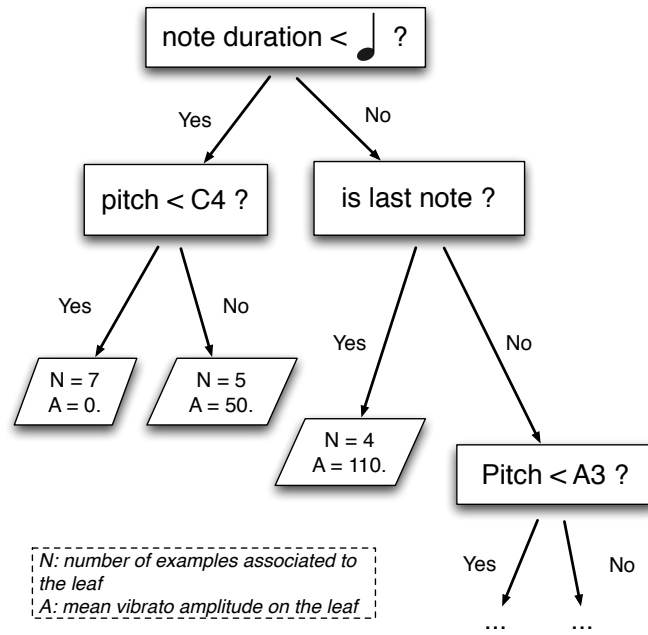
FIGURE 5.3: Illustration of a decision tree for a single parameter (vibrato amplitude)

...), it is useful to normalize them so that they lie in the same range and thus all have a similar impact on the choice of the best split. A possibility for this is to rescale the distribution of each parameter to have zero-mean and unit-variance.

A particular advantage of using decision trees is that they are "white box models", easy to understand and to interpret, as they can be visualized as a simple graph, contrarily to "black box models" such as neural networks. They can also handle both categorical and numerical data, which is advantageous, regarding the different natures of the contextual factors used, as will be detailed below. However, one should be aware of some possible drawbacks when using decision trees. A first possible problem is overfitting, which can be limited by setting appropriate stopping criteria. Note that there is however no absolute rule for choosing the best stopping criteria, which may especially depend on the quantity and consistency of the data available. Because the splitting process of the algorithm is only locally optimal, it also does not guarantee to return the globally optimal decision tree, and some small variations in the data may thus induce non-negligible changes in the structure of the tree in some cases. Finally, the trees might be biased if some particular contexts are more represented than others. It is thus important to have a well-balanced coverage of the contextual factors in the training dataset. In the next section, we discuss the various contextual factors that may be used for our purpose to build appropriate decision trees.

5.4.3 Contextual factors

A contextual factor can be defined as a variable partly representing the situation in which a certain sample of the features to be modeled (e.g. a parametric transition's template, or the duration of a specific consonant extracted from our corpus) has been observed. Examples of contextual factors are for instance the pitch or duration of a note, its position in a musical phrase, or the number of syllables in a

pronounced word. We denote in this work by the more generic term of "context" any combination of various contextual factors associated to a particular sample.

For statistical-based speech synthesis, rich sets of contexts have been defined in the literature, essentially related to the syntactic information, such as described in [SG11] or [Obi11]. However, singing being based on a musical score, different contextual factors must be considered. In [Our+10], many contextual factors have also been defined for singing, at different levels (song, phrase, note, japanese mora, and phoneme), for HMM-based synthesis. However, as previously evoked, other studies focusing more on expression control and singing style [STK10; Nos+15; UBB13a] tend to use more restricted sets of contextual factors, limited to the note or bar levels, and don't always consider the phonetic aspect. It seems important to us, though, to take into account a wider variety of contextual informations, such as the temporal or melodic position of a note in a whole musical phrase, or the phonemes pronounced, which may have influence on the interpretative choices of the singers. From the analysis of our corpus, we can for instance notice that Edith Piaf uses rather consistently a particularly present vibrato on the last note of musical phrases. In [IIO14b], the temporal position of a note in the musical phrase has been considered, but other contextual factors are missing. It was also shown in [LDL12] that the influence of the lyrics' phonetic on the f_0 is of importance and should thus be considered. Based on those previous studies and the experience of our musicologist collaborator Céline Chabot-Canet, we defined an extended list of all the potential contextual factors, at different levels, that may have to be considered in the context of singing style modeling, summarized in figure 5.4.

As shown in this figure, many different contextual factors may be considered. All of them may not be important for all styles or singers, but it is not always obvious to know which are really important and which are not. However, using the CART algorithm, using additional contextual factors of less importance is not a problem and won't degrade the results, as only the most influential ones will be selected by the algorithm when building the tree.
Those contextual factors relate to various aspects (rhythm, melody, syntactic and semantic features, ...) and structural levels of a song, encompassing the macro-structure of the song, the musical phrase, as well as very local levels like the note, the syllable or the phoneme. They are also of different natures, some of them being boolean, and other being continuous numerical values. However, note that using both discrete and continuous types of contextual factors is not a problem for the CART algorithm, as previously explained. Note that some contextual factors may sometimes be missing (e.g. the pitch of the next note, when the current note is the last of a musical phrase and thus followed by a silence). In such case, their value is simply set to -1 and treated as usual by the algorithm.

However, one should be aware that singing is not a fully deterministic process, and the expressive gestures of singers may be subject to some part of randomness or result from personal choices that might not always be predicted from those contextual informations.
Moreover, all contexts listed here are considered as potentially useful for some singing styles, but not all of them could be exploited in our work. This is the case for the macro-structure of the song, as we don't have enough songs in our corpus to properly cover those contexts. Although potentially important, the semantic informations could also not be exploited, because it would require to map
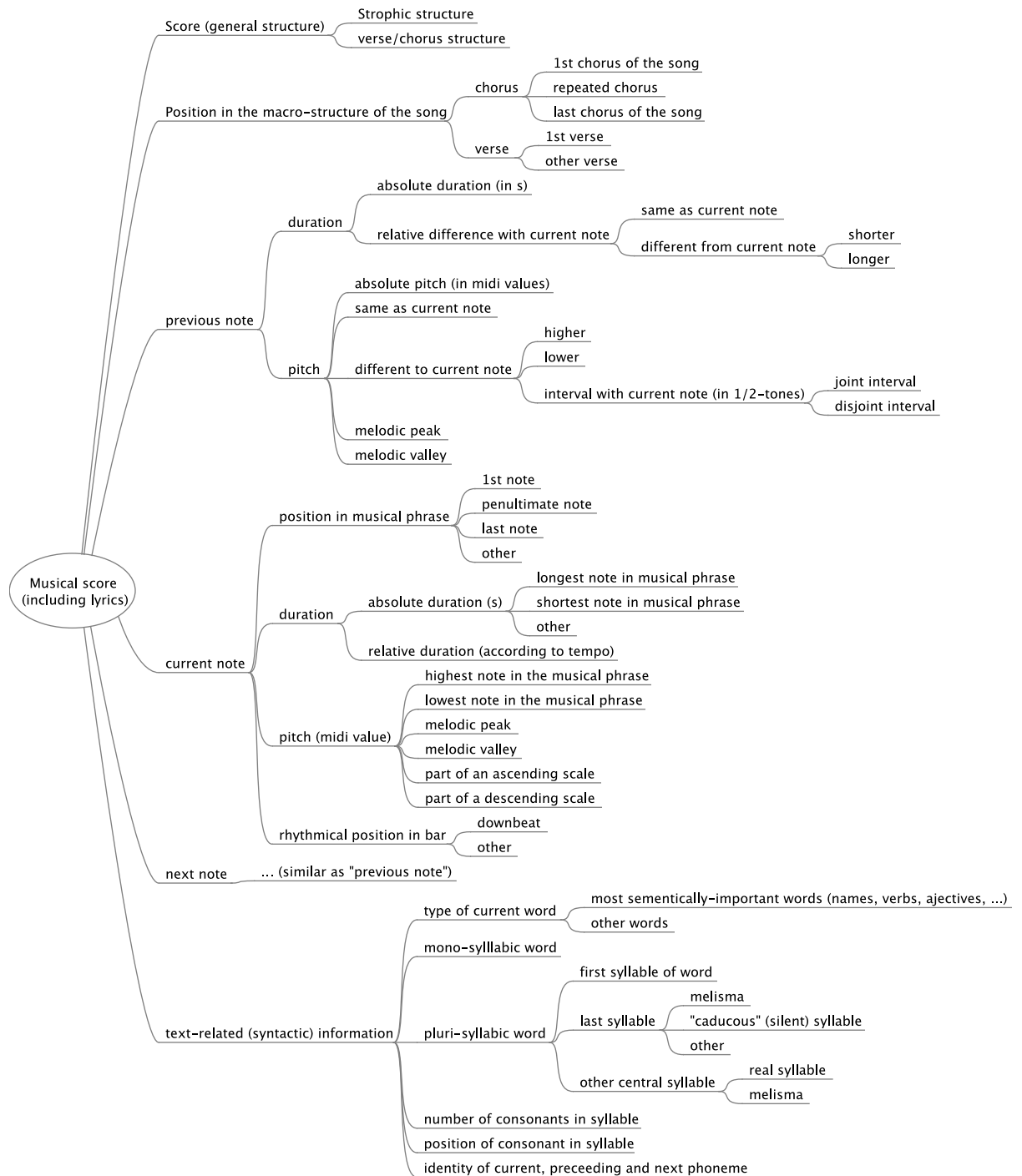
FIGURE 5.4: Structured list of all identified potential contextual factors (based on work from the musicologist Céline Chabot-Canet)

each phonemes in the input lyrics of the synthesis and in our corpus to the syllables and words to which they belong, in order to know those informations, which is not possible in the current state of our system and from the current annotations of our corpus. Similarly, the bars have not been annotated in our corpus, and the positions of the notes in the bar are thus unknown.
Regarding the other contextual factors, we define in this work a musical phrase by the set of notes comprised between 2 silences (although this definition is not always ideal, as small silences may sometimes be inserted inside of a musical phrase). The caducity of a syllable ("caducous syllable" in the figure) is a concept present in the French language, corresponding to a syllable at the end of a word finishing with a "silent" /@/ that is usually not pronounced in fluent speech, but that is pronounced in singing to support a note at the end of a musical phrase.

5.5 Estimation of parameters on the corpus

Once the corpus has been annotated, and previous to build styles models, it is necessary to extract from the corpus the f_0 and intensity parameters that will be used to build those styles models. For this purpose, our aim is to approximate as best as possible the real curves with the parametric models presented in the previous chapter.

5.5.1 fo model parameters

5.5.1.1 Pre-processing of the f_0 curve

Before extracting the f_0 model's parameter, a pre-processing step is applied on the f_0 analysis of the songs. Despite the manual correction of the curves, some spurious values may remain, which are discarded by setting a threshold on the confidence score returned by the f_0 estimation algorithm. Then, in order to limit the influence of the micro-prosodic variations on the estimation of parameters, the f_0 curve is linearly interpolated across each voiced consonants (except semi-vowels), based on the phonetic annotation, similarly to [STK10]. Finally each voiced segment of the curve is low-passed filtered using a hanning window, with -6dB bandwidth of 20Hz. Figure 5.5 shows an example comparing the curve before and after pre-processing. Then, all unvoiced consonants are also interpolated to obtain a continuous curve.

5.5.1.2 f_0 curve segmentation

As our f_0 model consists of a succession of segments with their own set of parameters, it is first necessary to segment the f_0 curve appropriately into the elementary units of the model (attacks, sustains, transitions, releases, and silences). In order to alleviate the need for manual work, this is done in a semi-automatic procedure, with manual correction. Such a semi-automatic procedure is also used in [Umb15] to delimit the transitions and sustains segments used by the proposed units selection and HMM-based systems, as well as a manual annotation for the first and last cycles of vibrato sections. In [MBL06], for the purpose of automatic evaluation of singing performances, the f_0 curve is also automatically segmented into attacks, sustains, transitions, releases and vibratos, using an algorithm based on untrained HMMs with probabilistic models built out of a set of heuristic rules. We present here the approach used to generate the first automatic pass of the curve
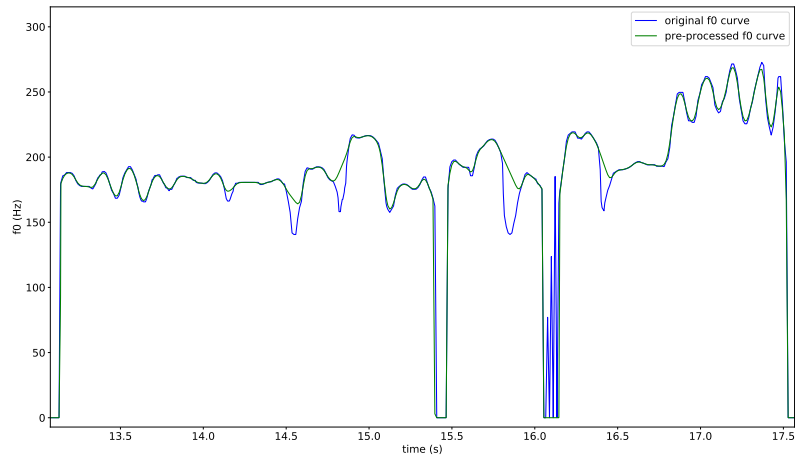
FIGURE 5.5: Comparison of an f_0 curve before and after pre-processing

segmentation, also based on heuristic rules.

Transitions' centers:
In our case, contrary to [Umb15] and [MBL06], we already have a notes segmentation in our corpus annotations. Then the first step in this automatic f_0 segmentation procedure is to find the center of transitions segments around notes' onset times. For each transition, this center position is searched on a segment spanning from the half of the left note to the half of the right note. In ideal cases, the transition's center should be simply positioned at the time of the maximum of the f_0 1^{st} derivative. However, the f_0 contours are not always as ideal and this unique condition is often not sufficient. For instance, important f_0 fluctuations due to vibrato may interfere in the choice of the right position, or some long transitions may have more complex profiles with several peaks in the derivative. To limit its influence, the vibrato is first reduced using a FIR low-pass filter (using a hanning window as filter's coefficients, whose length is equal to the supposed maximum vibrato cycle). Then, for a more robust detection, the transition's center t_c is assumed to lie at the maximum of a probability function, obtained by the multiplication of 4 weighting curves defined on the given f_0 segment around the note's onset:

$$t_c = \underset{t}{\operatorname{argmax}}(w(t)) = \underset{t}{\operatorname{argmax}}(w_1(t) \cdot w_2(t) \cdot w_3(t) \cdot w_4(t)) \tag{5.1}$$

The first weighting curve $w_1(t)$ is defined as the f_0 1^{st} derivative (multiplied by -1 for downward transitions, so that the negative minima of the derivative become positive maxima), normalized by its maximum value.
The 2^{nd} weighting curve $w_2(t)$ is equal to the concatenation of 2 half-hanning windows of the lenght of half the left and right notes respectively, which is used to favour positions closer to the note's onset.
Around the center of a transition, the absolute value of the 2^{nd} derivative is expected to be rather low. The 3^{rd} weighting curve $w_3(t)$ is thus defined by $w_3(t) = 1 - |f_0''(t)|$, where $f_0''(t)$ is the 2^{nd} derivative of the f_0, normalized by its maximum.
The last weighting curve $w_4(t)$ is used to favour positions around which the pitch
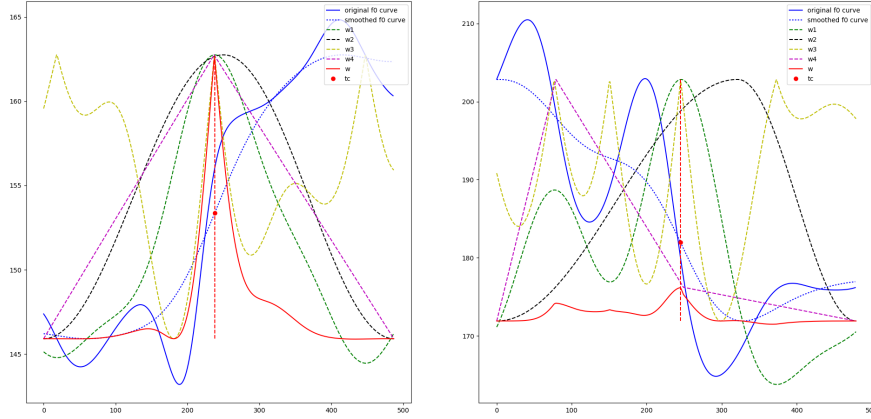
FIGURE 5.6: Examples of estimation of the transition's center for f_0 curve segmentation

change is as close as possible to the theoretical interval given by the midi annotation. For this purpose, a set of candidate positions is first extracted based on the local extrema of the 1st derivative (maxima for upward and same-note transitions, or minima for downward transitions). Then, for each candidate position, a value $\delta_i = \frac{1}{|\Delta_{p_i} - \hat{\Delta_{p_i}}|}$ is computed, where Δ_{p_i} is the expected pitch difference (based on the midi annotation), and $\hat{\Delta_{p_i}}$ is the actual pitch difference measured on the curve based on the closest f_0 extrema around the i^{th} candidate position. With this measure, the smaller the difference between the measured and expected interval, the bigger δ_i. Finally, $w_4(t)$ is obtained as a linear interpolation of the values δ_i between the candidates positions, and normalized by its maximum.
Note that the 4 weighting curves have been normalized to a maximum value of 1 to be of equal importance.
Figure 5.6 shows 2 examples of transitions with the original and smoothed f_0 curves, the 4 weighting curves, the final probability curve $w(t)$, and the chosen positions for the transition's center.

Transitions' boundaries:
Once the transitions' centers have been properly detected as described above, the transitions boundaries around those positions are found. The limits of a transition are assumed to be characterized either by a rather flat f_0 curve beyond the boundary (possibly before/after a preparation/overshoot), or by the start of a vibrato cycle (possibly directly chained with the preparation/overshoot).
Thus, for each transition, a segment is first defined, spanning 75% of both left and right notes, to limit the search of the boundaries. Then, on each side of the previously found transition's center, the flatness of the f_0 curve is verified. For this purpose, the f_0 is scanned in each direction, from the transition's center, and if the 1st and 2nd derivatives of the f_0 are both below a certain threshold for a minimum amount of time, the f_0 curve is assumed to be flat enough and the position at the beginning of the flat zone is memorized. The thresholds have been empirically set to 300 cents/s for the 1st derivative and to 100 for the 2nd derivative. The minimum time has been set to 0.05s.
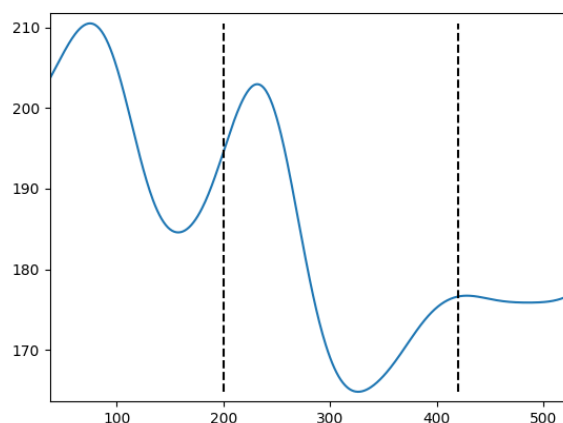
FIGURE 5.7: Example of transition's boundaries found with the automatic procedure

Then, the positions of the prominent peaks of the absolute value of the 1st derivative are found, which correspond to an f_0 variation in the opposite direction of the transition (as it should be the case at the beginning/end of a preparation/overshoot). The prominence of a peak is defined by the fact that the peak is surrounded by 2 minima and for which the difference between the peak value and both minima is superior to 10cents/s. Finally, for each side of the transition, if a flat zone has been detected, and there is not more than 1 prominent peak between the transition's center and the flat zone, the transition's boundary is set at the start of this zone. Otherwise, the last prominent peak on the left side, or first one for the right side, is retained as the boundary.
If the transition's boundaries don't encompass all consonants of the right syllable, they are corrected afterwards so that all consonants are contained in the transition. Once the boundaries have been found for all transitions, in case 2 transitions overlap, they are adjusted to the middle position of the 2 boundaries.
Figure 5.7 shows an example of transition's boundaries found with this technique. On the left, there is no flat zone because the transition is directly chained with a vibrato, and the boundary is thus set just before the preparation. On the right part of the figure, a flat zone is detected and the boundary is thus set just after the overshoot, at the beginning of the flat zone.

Attacks' boundaries:
For notes following a silence, the note starts with an attack segment. However, the attacks may possibly start on voiced consonants, before the onset of the vowel, but not necessarily at the start of the voiced consonant, as the consonants may be sustained at a lower pitch than the target pitch of the attacked note before the vowel's onset. The start of the attack is thus set at the position of the minimum f_0 value between the end of the silence and the vowel's onset (given by the phonetic annotation).
Similarly to the transitions, we then check for a flat zone at the right of the attack's start. If such a flat zone exists, then the end of the attack segment is set at the start of this zone. Otherwise, the end of the attack is set to the first time position where the derivative becomes negative or where the frequency goes above the median frequency of the note. (Note that the attacks may thus be as short as 1 sample if
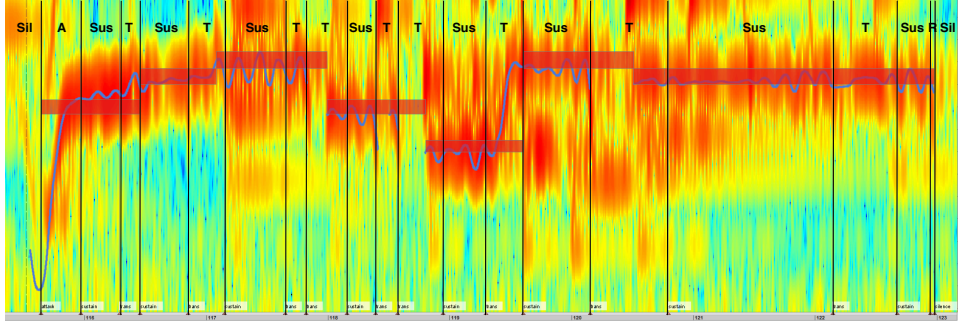
FIGURE 5.8: Example of automatic f_0 curve segmentation

the conditions are met.)

Releases' boundaries:
For release segments, before each silence, the same procedure than for attack segments is followed, in the opposite direction.

Sustains and silences:
Finally, sustain segments are placed between the transitions, attacks, and releases, and silence segments between each pair of release and attack.

Figure 5.8 shows an example of segmentation obtained with the procedure described, along with the f_0 curve and the midi notes, for an extract of "Les feuilles mortes" by François Le Roux. From this automatic segmentation, a marker file is generated and can be manually corrected (e.g. using audiosculpt), in order to obtain the most appropriate results for the parameters estimation.

Once this segmentation is established, the parameters of each segment, previously illustrated in figure 4.8, can be estimated as will be now explained in the following section.

5.5.1.3 Estimation of segments' parameters

Sustains segments:
For sustains segments, the vibrato is first removed by interpolating between the local extrema of the f_0 derivative (between each half vibrato cycle) to obtain a smooth curve crossing the vibrato cycles, and further low-pass filtering the result using a FIR filter whose coefficients are made of a hanning window with a size of twice the maximum vibrato cycle, similarly to [RM11]. An example is shown in figure 5.9. Then, this curve is subtracted from the original one to obtain only the frequency modulation related to the vibrato, centered around 0 (as shown in figure 5.11), and the extrema of this new curve are found. In case several succeeding extrema with the same sign are found, only the one with the biggest absolute value is kept (black dots in figure 5.10).
In case the duration of the sustain segment is shorter than the minimal possible duration of a vibrato cycle, or if the biggest absolute value of the extrema corresponds to an amplitude inferior to a minimum threshold of 10 cents, it is assumed that there is no vibrato. Otherwise, the vibrato periods are computed according to the distance between the extrema, and a local vibrato frequency is obtained for each half-cycle of vibrato. For long notes, this is done using only the
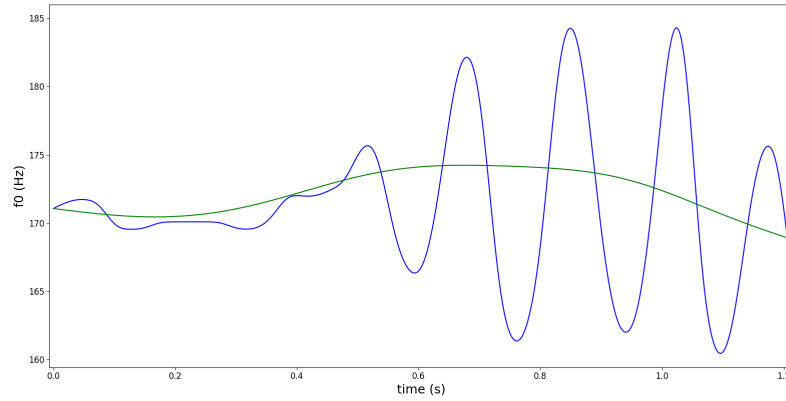
FIGURE 5.9: Example of vibrato removal on a sustain segment

central part (2/3 of the extrema), where the vibrato is assumed to be the most stable. Finally, the median of those values is chosen as the final vibrato frequency f_{vib}.

Then, a continuous vibrato amplitude envelope is obtained by linearly interpolating the extrema values, to which the previously-described piecewise-linear ASR amplitude profile (including a possible offset time before the start of the vibrato) is fitted, using a grid-search procedure. An appropriate grid is defined in order to reduce the search space and thus reduce the computation time on long notes. For the vibrato amplitude A_{vib}, a step size of 10 cents is used between the smallest and the biggest absolute values at extrema positions. For the offset time T_o, a step size equal to half a vibrato cycle is used, from 0 up to the position of the first extrema above 20 cents. For the attack time T_a, the maximum value is set to the position of the maximal amplitude (biggest extrema), and a step size of 25ms is used. Similarly, for the release time T_r a step size of 25ms is used between the biggest extrema and the end of the sustain segment. Then, the parameters are estimated using a brute-force approach on this grid, with the Mean Squared Error as error function (computed between the generated ASR envelope and the real one). In case $T_o + T_a + T_r > d_{seg}$, where d_{seg} is the duration of the sustain segment, the error value is set to infinity. Figure 5.10 shows an example of such an ASR amplitude curve fitted on a vibrato.

Transitions segments:
For the transition segments, the durations of the left and right parts d_L and d_R are computed according to the segment's boundary and transition's center position. The amplitudes of the preparation and overshoot A_L and A_R are computed as the distance, in cents, between the extrema frequency f_{min} (respectively f_{max}), on each side of the transition, and the frequency f_1 (respectively f_2) at the segment's boundary, as already explained in section 4.4.2.1: $A_L = 1200 \cdot log2(\frac{f_{min}}{f_1})$ et $AR = 1200 \cdot log2(\frac{f_{max}}{f_2})$.

Attacks and releases segments:
For attacks and releases, D is computed as the distance, in cents, between the minimum value f_{min}, and the final (respectively initial) value f_{end} (resp. f_{start}) of the f_0 segment: $D = 1200 \cdot log2(\frac{f_{min}}{f_{end}})$. The length L is obtained directly from the
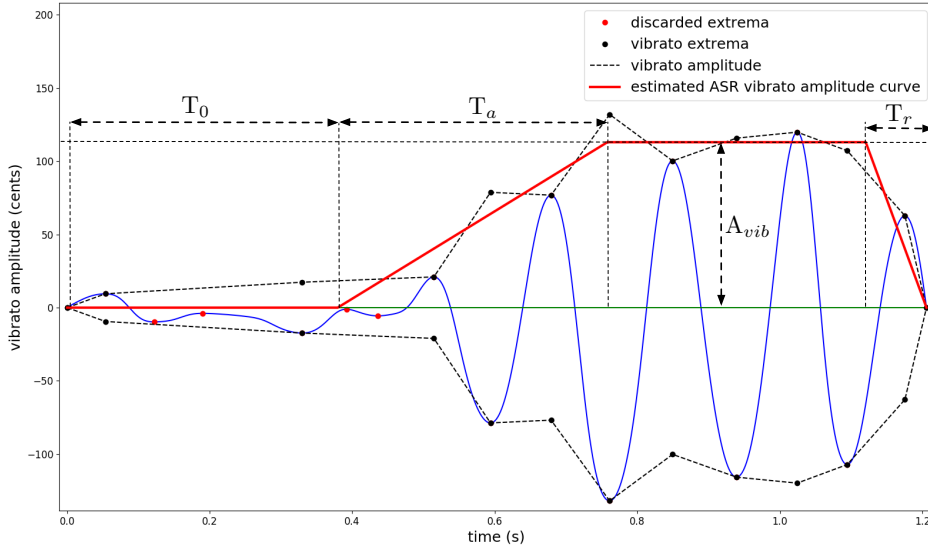
FIGURE 5.10: Vibrato parameters estimation. The vibrato is centered around 0. Then extrema are found for each half cycle. Finally, a continuous amplitude envelope is obtained by linearly interpolating the extrema, to which the ASR envelope is fit.

segment's boundaries.

Once the parameters have been estimated for each segment, the f_0 curve can be reconstructed from the model using those parameters. Figure 5.11 compares an extract of a real f_0 curve with the synthetic curve generated from the estimated parameters, for the same extract as figure 5.8. As one can see on the figure, the reproduction is not perfect, but the synthesized curve is rather close to the original one, and most of the expressive features have been captured (vibrato amplitudes and frequencies, attack's depth, transitions' shapes, ...). When synthesized using pure sinusoids, very few differences can be heard between those 2 curves. Example of such resynthesis of the original and synthetic f_0 curves are given respectively in sound 5.4 and sound 5.5. (Note that the frequency has been set to 0 on the figure for unvoiced parts.)

5.5.2 Intensity model parameters

As explained in section 4.5, the intensity is parametrized using a piece-wise linear ASR envelope for each note, that is to be fitted on the real loudness curve only on the vowels segments. For this purpose, the loudness curve is first smoothed using a low-pass filter. There is however no absolute reference to compare the loudness curves between the recordings (which depend on the gains used), and we are thus only interested in the relative variations. For this reason, each curve is first normalized by its maximum value over the whole song. Then, the intensity model being similar to that used for the vibrato amplitude envelope, the same grid-search procedure is used to optimize the parameters, using the MSE between the generated amplitude envelope and the real contour as error function. The step δ_t used for the attack and the release time is $\delta_t = max(0.05, \frac{d}{10})$ where d is the duration of the vowel. For the global amplitude of the segment, a step of 0.05 between the minimum and maximum values is used. For the attack's and release's depths, parametrized as a ratio of the global amplitude, the step used is 0.05, with
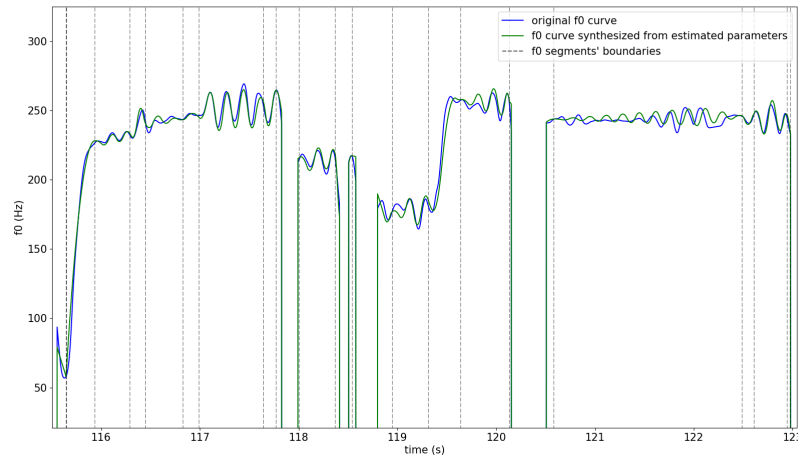
FIGURE 5.11: Example of f_0 synthesized from the estimated parameters against the real f_0 curve
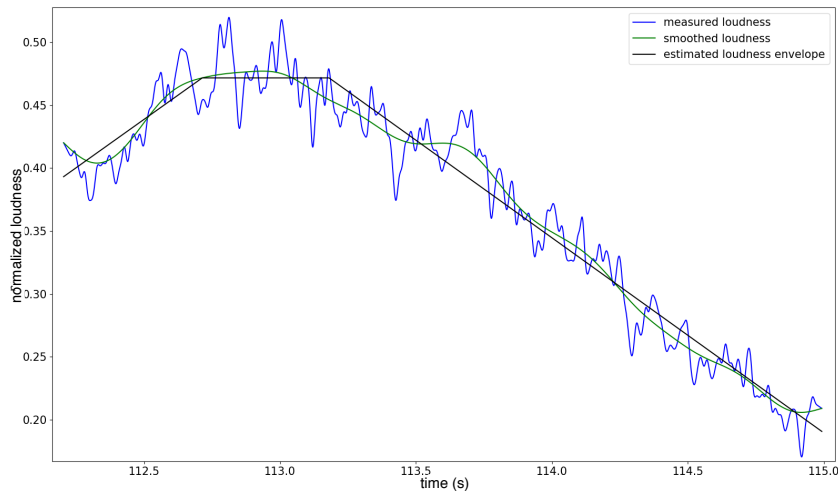
FIGURE 5.12: Example of estimated loudness profile for 1 note, fitting an ASR envelope on the real curve

a minimum of 0 and a maximum $detph_{max} = 1 - \frac{l_{min}}{l_{max}}$, where l_{min} and l_{max} are the minimum and maximum values of the loudness segment. An example of an ASR envelope fitted on a real loudness segment (from a single note) is shown in figure 5.12.

Figure 5.13 show this estimation on a longer extract. The vertical lines represent the limits of vowels and loudness has been set to 0 on consonants for clarity. As one can see, the proposed approach gives a reasonable approximation of the measured loudness contours. Note that for the 8th note, the contour may be better fitted by using an offset time before the start of the crescendo, but this has not been included for simplicity, assuming that this case is quite rare and that the perceived difference is not too important.

Besides the loudness profile of each individual note, the mean loudness over each musical phrase (between 2 silences) is also extracted, as well as the ratio between the maximum loudness of each note and that of the previous one, which are required for the prediction of the loudness curve at synthesis, as will be explained
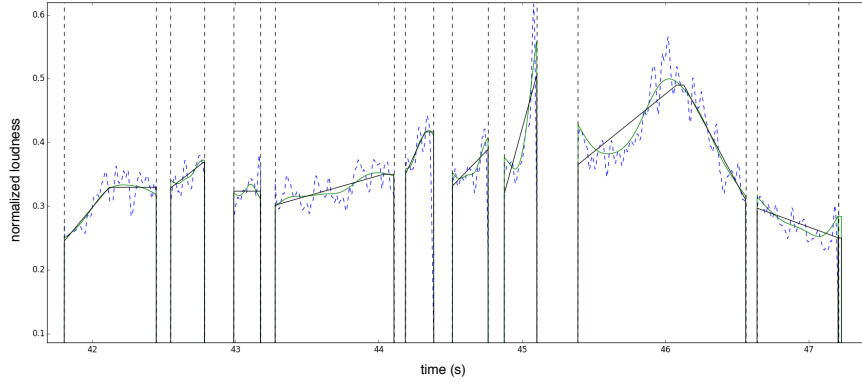
FIGURE 5.13: Loudness model fitted on each note of a song extract

in section 5.6.3.

Along with those parameters, we also extract the corresponding contextual factors to constitute our styles databases, as illustrated in figure 5.2 for f_0 transitions. From these data, we can then build decision trees to constitute the style models.

5.5.3 Comparison of parameters between styles

Based on those parameters estimations, one can observe and compare the distributions obtained for each style. Due to the high number of parameters and the many contextual factor that may influence their values, a detailed analysis of the estimated parameters would be very complex and fastidious. Nevertheless, we wish to give here some insight on the differences between the styles, based on a few global observations that tend to correlate with the general description of the styles given in section 5.3.2. Figure 5.14 shows the distributions of the vibrato amplitudes on all the notes longer than 0.5s for the 4 styles of our corpus. The main observation is that François Le Roux tends to use a much wider vibrato than other singers, mostly around 100cents, and up to 200, although the amplitude seems to vary a lot. Distel and Greco are the ones with the smaller vibrato amplitudes with the bigger peak under 50cents.

Figure 5.15 shows the distributions of the attacks' depths for notes longer than 0.25 seconds. It is clear from this figure that François Le Roux uses much smaller attacks than other singers.

Figure 5.16 shows the distribution of the preparations' amplitudes in upward transitions when both the left and right notes are longer than 0.25s.

Figure 5.17 shows the distributions of the duration, in seconds, of the phoneme /R/ for the 4 styles on notes longer than 0.5s. As one can see, Edith Piaf and Juliette Greco tend to use longer /R/ than the others. Note that in this plot, the note's duration is long enough (0.5s) so that the phoneme's duration is not constrained by the note's duration.

Figure 5.18 shows the distribution of the attacks' depths of the intensity model for notes longer than 1s.

Although we only plotted here the values of a few parameters in certain contexts, this already shows some differences between the styles that have been captured by the proposed parametrization and that should thus be reflected in the
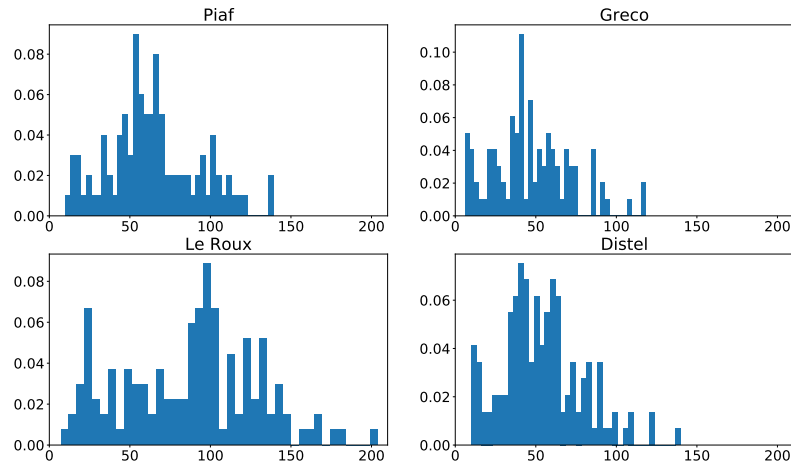
FIGURE 5.14: Distributions of the vibrato amplitudes (in cents) for the 4 singers of our corpus
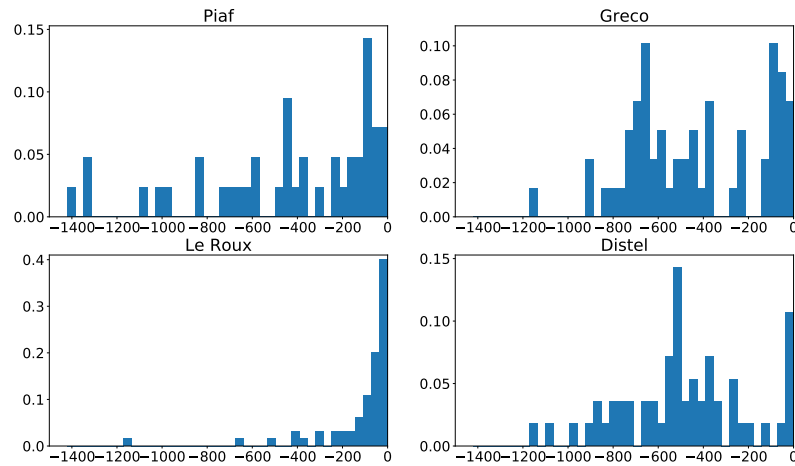
FIGURE 5.15: Distribution of attacks' depths (in cents) for the 4 singers of our corpus

synthesis when modeling those different styles.

We now detail in the following sections the contextual factors and specific procedures that have been used for generating style-dependant models to be used during the synthesis for each parameter.

5.6 Styles models and parameters generation

5.6.1 Phonemes durations

As said in section 4.3.2, consonants' durations vary in a non-linear way with tempo and context and may be used by singers as an expressive mean to purposely accentuate certain notes. However, it seems that this aspect has not been much considered
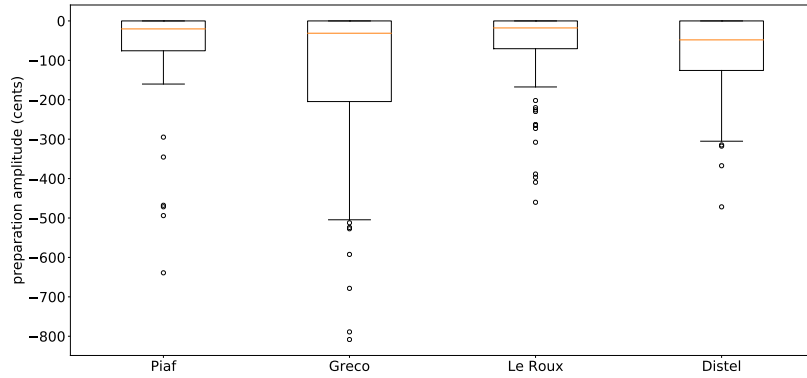
FIGURE 5.16: Distribution of the preparations' amplitudes in upward transitions (in case the left and right notes are longer than 0.25s)
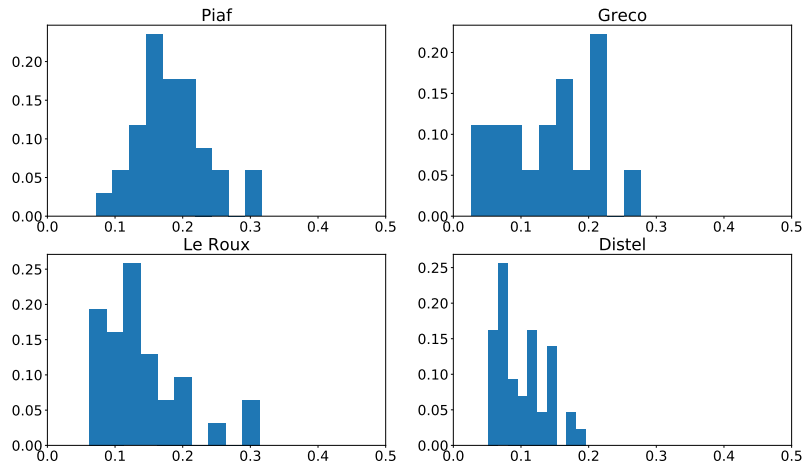
FIGURE 5.17: Distributions of the duration of the consonant /R/ (in seconds)

in previous works on singing style modeling. We thus propose here to include this aspect in our styles models.
For this purpose, the durations are directly extracted from the phonetic segmentations of the corpus. As some phonemes may not be encountered in a wide variety of contexts due to the small size of our corpus, phonemes are grouped into phonetic classes with other phonemes having similar articulatory characteristics (and thus hopefully similar durations) for a better coverage of contexts. Then during the learning stage, a decision tree is built for each phonetic class. The phonetic classes that we have used are (with the corresponding phonemes in SAMPA notation): voiced fricatives (/v/,/z/,/Z/); unvoiced fricatives (/f/,/s/,/S/); voiced plosives (/b/,/d/,/g/); unvoiced plosives (/p/,/t/,/k/); nasals (/m/,/n/, /N/); semi-vowels (/w/,/j/,/H/); /R/; and /l/. The phoneme's identity then becomes itself a contextual factor used in the tree building for each of those classes.
The contextual factors used for the tree building mainly encompass:
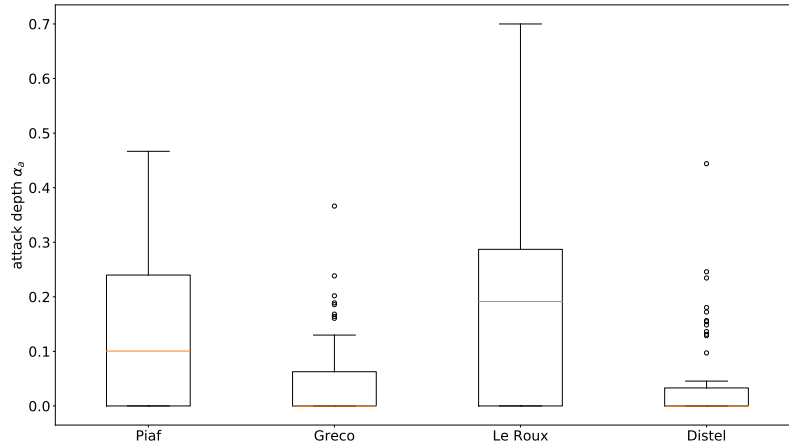
FIGURE 5.18: Distribution of attack's depth α_a of intensity model for notes longer than 1s.

– the class and identities of previous, current and next phonemes;
– the number of successive consonants, and position of the current one;
– the duration of the previous consonant if any;
– the pitch and durations of the previous, current and next notes;
– the durations and pitch differences with the previous and next notes;
– the temporal positions of the current note in the musical phrase (first, penultimate, or last note);
– the melodic positions of the previous, current and next notes in the musical sentence (ascending or descending scale, melodic peak or valley, highest or lowest note in sentence);
– the caducity of the current or next note;

The full detailed list of the contextual factors used is detailed in annexe, in appendix in section C.1. Note however that they are not all present in the final trees, depending on the output of the CART algorithm. As the tempo varies between songs, it seems more reliable for the contextual factors related to notes durations to use only the absolute durations in s, rather than symbolic durations relative to the tempo.
The stopping criteria used for building those trees is to have at least 5% of all samples on each leaf, with a minimum of 5. Figure 5.19 shows an example of a tree for the phoneme /R/ of the Greco style model. for each node, the question asked, the MSE, the number of samples and the mean phonemes duration (in s) on the node are given. The color varies according to the range of the predicted value (the bigger the darker). As one can see, for longer notes (on the right side of the tree), the phoneme's duration tends to be much higher than for the shorter notes (on the left side of the tree). But although the note's duration seems to be the most important factor (as could be expected), many other contextual factors are also used.

During the synthesis stage, consonants durations being used as contextual factors for the f_0 modeling, they need to be fixed first. For each consonant in the input phonetized text, the procedure previously explained in section 5.4.1 is used, reading
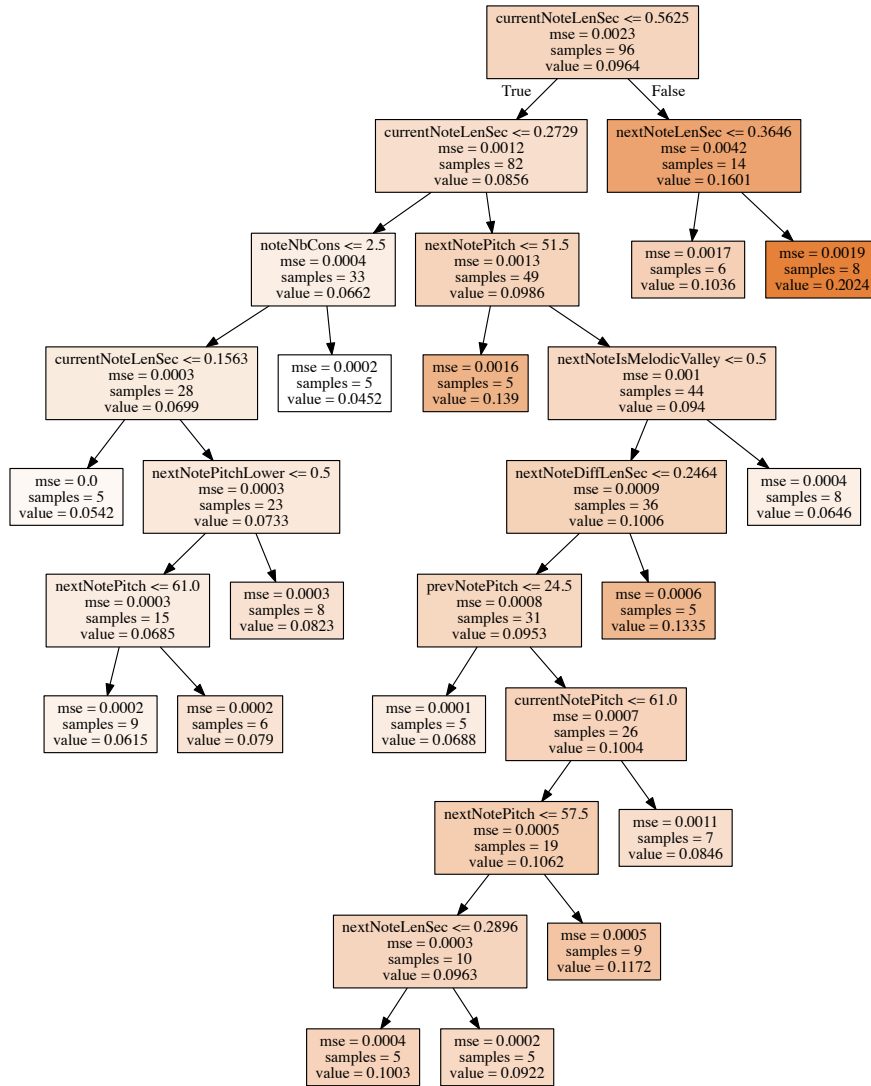
FIGURE 5.19: Example of a decision tree built for the phoneme /R/ of the Greco style model

the values that correspond to the target context from the tree of the corresponding phonetic class.

5.6.2 f_0

At synthesis stage, once the consonants' durations have been fixed, the sequence of segments of the f_0 model is first determined according to the notes in the score, as explained in the previous chapter, section 4.4.2.3. Then, for each segment, the same procedure as for the phonemes' durations is used to select a set of parameters among the samples from the database that are associated to the target leaf of a decision tree.
As already said, rather than a statistical modelization of those parameters, our approach is to select parametric templates, where the parameters for each template are not considered independently, but tied together. For this purpose, we thus used multi-output decision trees [Bor+15]. Such a tree is built for each type of segment of the model. However, we differentiate transitions according to the pitch interval direction (ascending, descending, or same-note), similarly to [Umb15], to avoid for instance to select a downward transition in place of an upward transition at synthesis. But as our corpus is made of real singing with lyrics, contrarily to [Umb15] which used recordings with only vowels, only the voiced transitions extracted from the corpus are used. A total of 6 decision trees are thus built for the different segments of our f_0 model: 1 for the sustains, 1 for the attacks, 1 for the releases, and 3 for the transitions. This distinction is rather similar to what is done in the 2^{nd} HMM-based approach proposed in [Umb15], where 6 different models are also built for sustains, attacks, releases, and the 3 types of transitions.
For sustain segments, the estimated parameters are the amplitude, offset time, attack time and release time of the vibrato. The vibrato frequency being assumed to be not controllable and rather stable for each singer, it is not considered in the tree building, and we use as vibrato frequency for each singer only the median value over all the sustained notes that are longer than 0.2s and that carry a vibrato. This avoids choosing undesirable and unexpected vibrato frequencies, which may otherwise occur sometimes due to possible errors in the f_0 analysis and vibrato frequency estimation. Figure 5.20 shows the distributions of the vibrato frequency estimated on the corpus for each of the 4 styles. For transitions, the estimated parameters are the durations of the left and right parts, and the amplitudes of the preparation and overshoot. For attacks and releases, the parameters are the depth and duration.

Model parameters having different ranges and dimensions (lengths in s and amplitudes in cents), they are first normalized previous to building the tree so that they all lie in the same range. The normalization is done by removing the mean and dividing by the standard deviation for each parameter, so that the distribution has zero-mean and unit-variance. Note that in a first publication on this approach [ACR16b], the parameters were normalized by their maximum. However, this may be more sensitive to the influence of possible outliers with high values, and using the mean and variance thus seems better-suited.
Moreover, different weights can then be set on each parameter so that some parameters judged as perceptually more important have more impact on the tree building. Table 5.2 summarizes the weights used for each parameter. We assume that using such weights can help to build better trees by putting more emphasis on parameters that seem perceptually more important. For vibrato, we assume that the most perceptually important aspect is its amplitude, as a long attack or release time is not
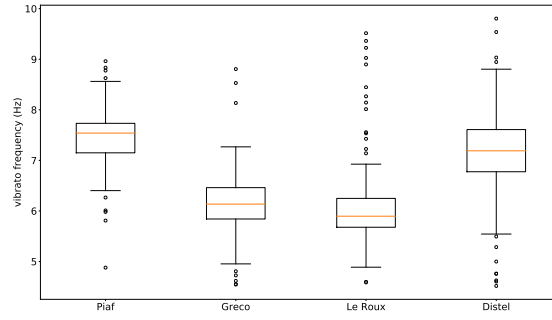
FIGURE 5.20: Distributions of the vibrato frequencies estimated on the corpus

vibrato		upward and downard transitions		same-note transitions		attacks		releases	
A_{vib}	3	d_L	2	d_L	1	D	2	D	2
T_0	2	d_R	2	d_R	1	l	1	l	1
T_a	1	A_L	1	A_L	2				
T_r	1	A_R	1	A_R	0				

TABLE 5.2: Weights of f_0 parameters used for building decision trees

much probable for vibrato of small amplitudes. The offset time T_0, by delaying the vibrato, put more emphasis towards the end of the note, which is also an important expressive effect to be considered. For upward and downward transitions, we assume that the durations are more important for the prediction, as the preparation and overshoot's amplitudes are rather dependant on those durations (it is not very likely for instance to have a very deep preparation if the transition's duration is very short). However, for same-note transitions, which are characterized by a possible f_0 valley, the most important parameter is the preparation's amplitude which creates this valley. We assume that their is no overshoot for same-note transitions and thus give it a weight of 0. Finally, for attacks and releases, the depth is perceptually more important than the duration, as for instance a long attack with a very small depth won't do much difference compare to a flat sustain segment, but the attack will be well perceived if the depth is important, whatever the duration.

The contexts used for building the trees are for the main part rather similar to those used for modeling the phonemes durations, but with some differences to take into account the specificities of each segment. As vibrato is only carried by the vowels, the other phonemes are not considered in the contexts. Transitions being at the junction of 2 notes, only 2 notes (left and right) are considered in the contexts. The presence of certain voiced phonemes and their duration is also considered, as we assume they may influence the duration of the transition, as well as the preparation or overshoot's amplitude. Attacks and releases being respectively at the begining and end of a sentence, just after or before a silence, only 1 note is considered in the contexts. The detailed list of contextual factors used for each type of segment is detailed in appendix, in section C.2.

The stopping criteria used for building those trees has been empirically set to

a minimum of 1%, and more than 2 samples, on each leaf of the tree. Figure 5.21 shows the first levels of the decision tree built for upward transitions of the Greco style. For each node, the question asked, the number of samples, the MSE and the mean values of the normalized parameters are given. In this figure, the color varies according to the purity of the node (the darker the purer).
Figures 5.22, 5.23 and 5.24 show respectively all the upward voice transitions from our corpus for the Greco style, and the transitions contained in the nodes A and B from figure 5.21. In the plot, the value 0 on the time axis corresponds to the center of the transition, and all transitions have been normalized in frequency, between 0 and 1, by subtracting their minimum frequency and dividing by their ambitus. As can be seen in those figures, there is a rather high disparity in the durations and shapes when considering all the transitions together, but the decision tree managed to cluster appropriately transitions with similar durations and shapes, based on the provided contexts.

Once the parameters have been chosen for each segment, the specific additional rules and corrections detailed in the previous chapter, sections 4.4.2.6 and 4.4.2.7, are used in case of overlapping segments and for a correct placement of transitions according to the phonemes' timing. Some limits may also be used to constrain the parameters to lie in a given range (e.g. to avoid too important overshoots), which can be useful for discarding possible outliers due to errors in the annotation or in the parameters' extraction previous to the construction of the trees.

5.6.3 Intensity

For generating the intensity curve, the mean normalized loudness (between 0 and 1) must first be given for each musical phrase to be synthesized, a value of 1 corresponding to a maximum loudness level to be defined by the user. Ideally, this value should be automatically inferred according to the high-level structure of the song, but this would require much more data to be properly modeled. This aspect is thus not considered in our system and this value is left to the choice of the user.

Then, knowing this mean loudness value, our aim is to generate a loudness profile for each note in the musical phrase. Contrary to the f_0 for which the pitch is given in the score for each note, we assume here that the intensity of each note is not given in the score and should thus be automatically generated. Moreover, we assume that the intensity of a note should be included in the contextual factors for generating the possible crescendo and decrescendo on each note (a note with a high maximum intensity is more likely to carry a crescendo than a note with a low intensity). We thus decompose the loudness curve generation in 2 steps. First, a "static" loudness value (I_{max} according to the notations in figure 4.22) is determined for each note, without considering the dynamic variations (crescendo, decrescendo). Then, once this maximum loudness value I_{max} is set for each note, the dynamic parameters (d_a, α_a, d_r, and α_r, shown in figure 4.22) can be generated. For this purpose, 2 decision trees are thus built: 1 for predicting the static loudness value I_{max}, and 1 for predicting the other dynamic parameters in the second step.

As already said, we are only interested here in predicting the relative loudness values, as the absolute value depends on the gain of the recording and should thus be set by the user to obtain a desired level. The difference of loudness between
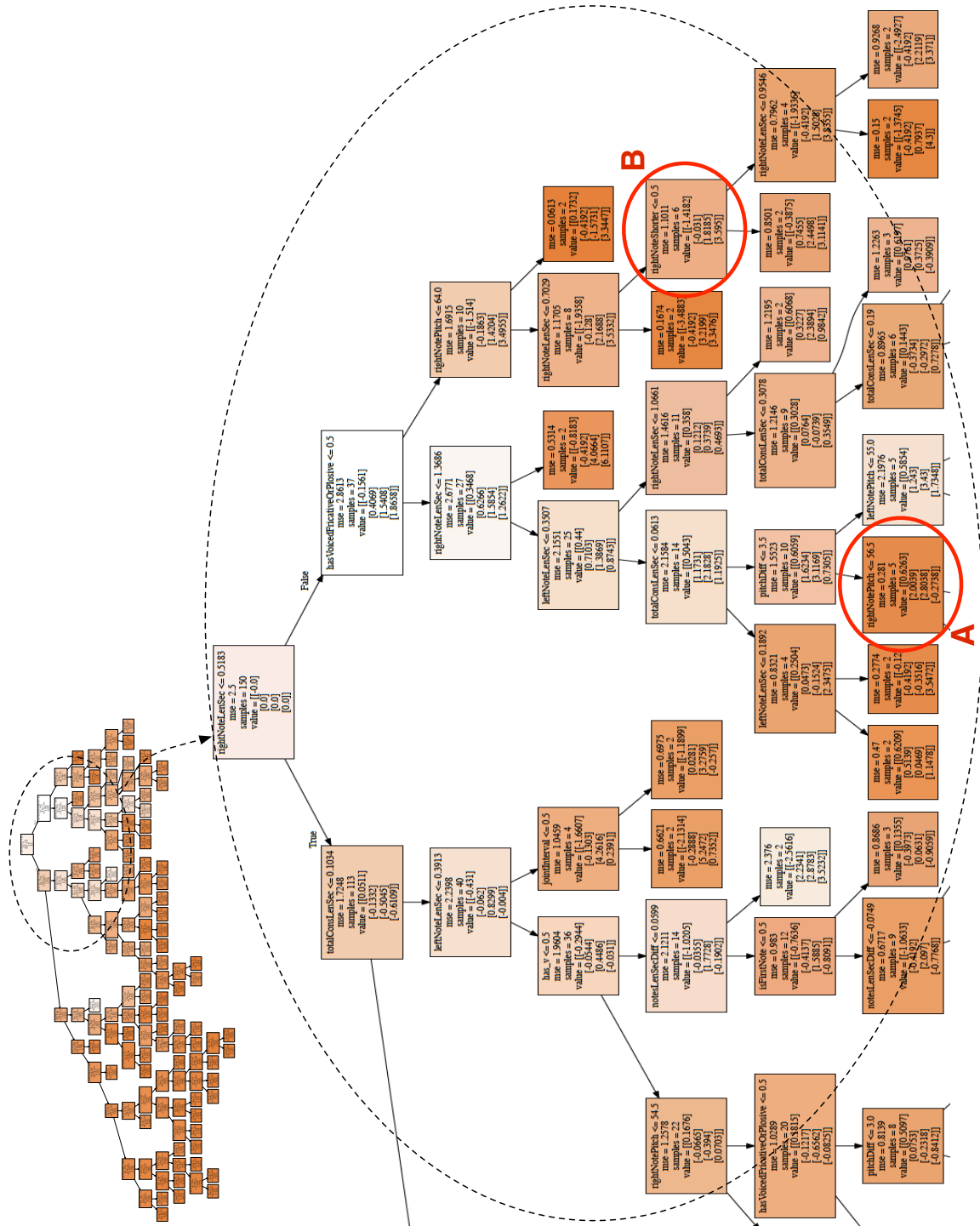
FIGURE 5.21: Example of decision tree for upward transitions of the Greco style model. The transitions contained in nodes A and B are plotted in figures 5.23 and 5.24 below.
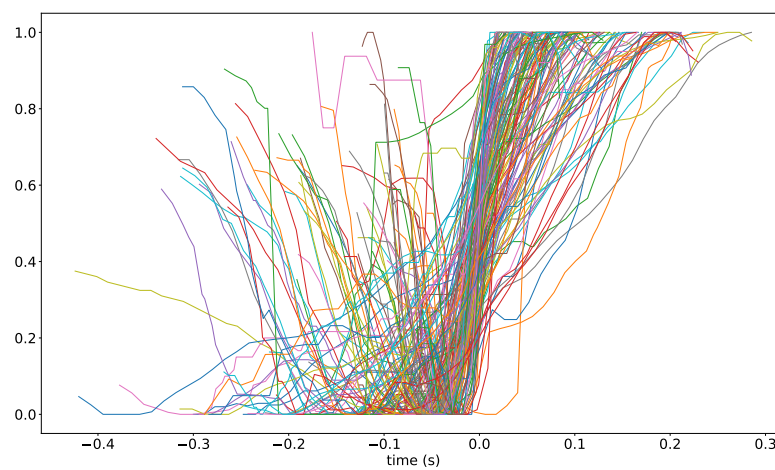
FIGURE 5.22: All upward voiced transitions for the Greco style
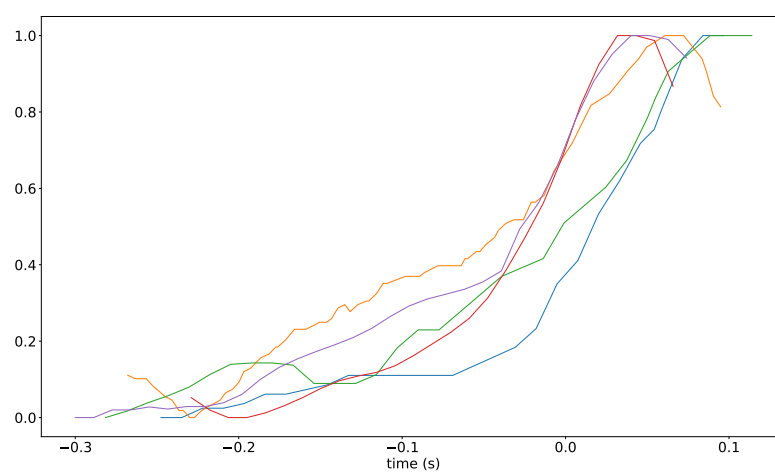
FIGURE 5.23: Transitions contained in node A from figure 5.21
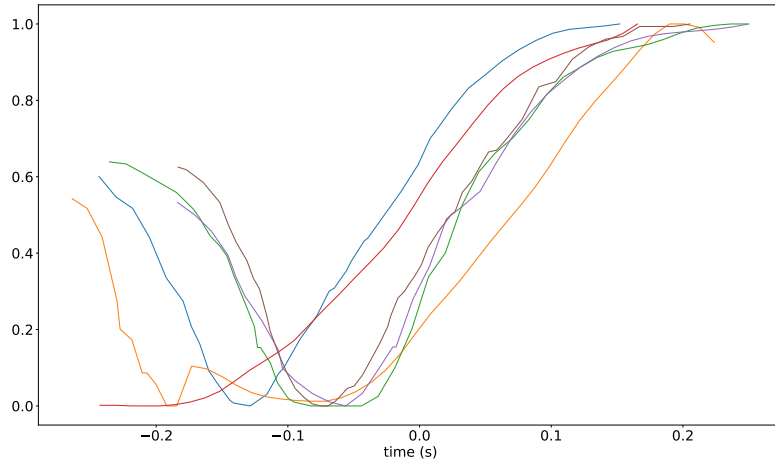
FIGURE 5.24: Transitions contained in node B from figure 5.21
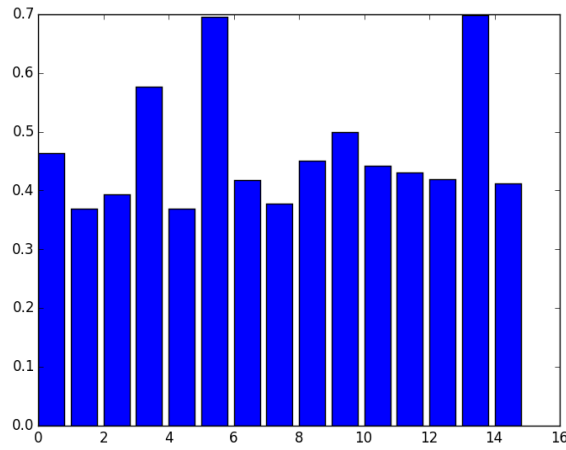
FIGURE 5.25: Example of loudness value I_{max} for each note of a musical phrase, for a direct prediction of I_{max}

values 0.25 and 0.5 and between values 0.5 and 1 should thus be considered to be equivalent when computing the mean square error used for building the tree, as in both cases the ratio is 2. For this purpose, the tree is built using $\log_2(I_{max})$ as target feature to be fitted instead of I_{max}.
Then, our initial approach was to directly use this tree to predict I_{max} according to the context. But this first approach considers each note independently without taking into account the note-to-note variations. The consequence is that in some cases, the difference of loudness obtained from one note to the next could be too important. Figure 5.25 shows as an example the predicted loudness values for all the notes in one musical phrase using this first strategy. As one can see, there are some rather abrupt changes on notes 5 and 14, which are about twice as loud as the surrounding ones.

To avoid this, we tried to rather predict the evolution of the loudness over time as the ratio of loudness from one note to the next, also extracted from the corpus. By fixing the level on the first note (e.g. using the direct prediction of the first
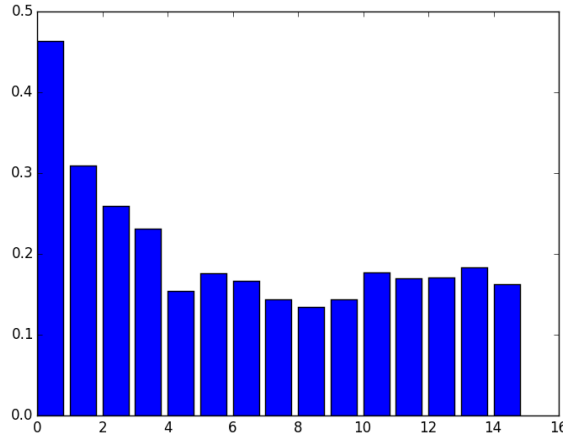
FIGURE 5.26: Example of loudness value I_{max} for each note of a musical phrase, based on the prediction of the note-to-note loudness ratio $r_{I_{max}}$

approach), the level of each note is then set according to that of the previous one based on the predicted ratio, which should thus avoid too abrupt variations that are not encountered in the corpus. In this case, we also use the log_2 value of the ratio, for the same reason as evoked above. But then, another problem may arise: in case the predicted ratio is quite monotonous, the change of loudness can go in the same direction for several consecutive notes, and this may rapidly lead to either very high or very weak loudness values. Figure 5.26 shows a possible result with this 2^{nd} strategy. As one can see, the loudness goes very rapidly down to a small value.

In order to limit the problems of those 2 strategies, it appears necessary to take into account both the possible loudness values on each note (according to the mean value of the sentence), and its note-to-note variations. Our final approach has thus been to simultaneously predict both the loudness values I_{max} and the associated loudness ratio with the preceding note $r_{I_{max}}$, and find the best compromise between the two. For this purpose, a multi-target regression tree is thus built for fitting both $log_2(I_{max})$ and $log_2(r_{I_{max}})$. The same normalization by the mean and variance as for the f_0 trees is also used here. Then for each note i in the sentence, both values I^i_{max} and $r^i_{I_{max}}$ are predicted simultaneously from this tree. The contexts used to build this tree are listed in appendix, section C.3. Finally, we try to find the best compromise $\hat{I}$ between the 2 sequences by minimizing the sum of 2 error functions, as described in the following equations:

$$\hat{I} = \underset{I}{\operatorname{argmin}} \sum_{i=1}^{N} (\epsilon_I(I) + \epsilon_R(I))^2 \tag{5.2}$$

with

$$\epsilon_I(I) = \left|\log_2(\frac{I}{I^0})\right| = \left|\log_2(I) - \log_2(I^0)\right| \tag{5.3}$$

$$\epsilon_R(I) = \left|\log_2(R(I)) - \log_2(R^0)\right| \tag{5.4}$$

where $I = [I_{max,0}, ..., I_{max,N}]$ is the predicted sequence of loudness values to be optimized, $R(I) = [r_1, ..., r_N]$ is the sequence of loudness ratios to be optimized,
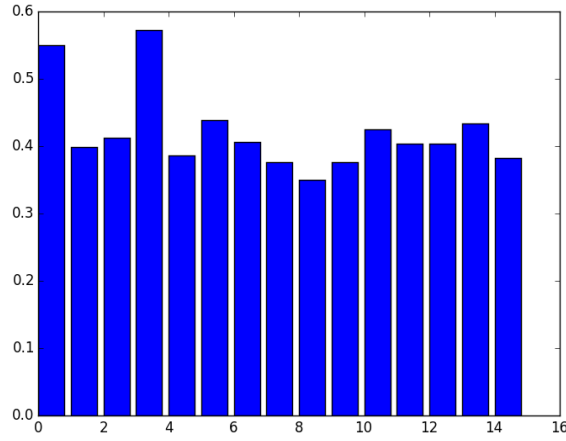
FIGURE 5.27: Example of optimized loudness value I_{max} for each note of a musical phrase, based on the described approach

with $r_i = \frac{I_{max,i}}{I_{max,i-1}}$, and N is the number of notes in the musical phrase. I^0 and R^0 are the sequences of loudness and ratio values predicted from the decision tree.

The minimization of this error function is then obtained using a modified version of the levenberg-marquardt [Mor78] algorithm implemented in the leastsq function of the scipy.optimize python package[6][7]. The algorithm is initialized with the mean of the 2 sequences generated by the 2 independent approaches described above, based on the values obtained from the decision tree. An example of result of this approach, for the same example as the previous figures, is shown in figure 5.27.

Another strategy may be to use the viterbi algorithm to choose the best path among the possible values, using a cost function based on the ratios of loudness with the surrounding notes so to better model the loudness variations over time. But this possibility has not been tested yet.

Once the final sequence has been determined, a factor is applied on all values $I_{max,i}$ so that the mean loudness value over the whole musical phrase corresponds to that given by the user.
Then, in a second step, the dynamic variations can be estimated. For this purpose, a second multi-target decision tree is built for parameters d_a, α_a, d_r, and α_r. The contextual factors used are the same as for the 1st tree, to which are added the loudness value I_{max} and the ratio of loudness with the previous note $r_{I_{max}}$. The values are also normalized as for the other trees.

Finally the generated normalized loudness curve is rescaled according to a given maximum value to have an appropriate absolute level. Figure 5.28 shows an example of a curve generated for the beginning of "Les feuilles mortes" in the Piaf style (note that only the vowels segments are shown here for clarity, but the curve should be linearly interpolated in-between). Based on this target curve, the

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.leastsq.html
[7] http://www.netlib.org/minpack/lmdif.f
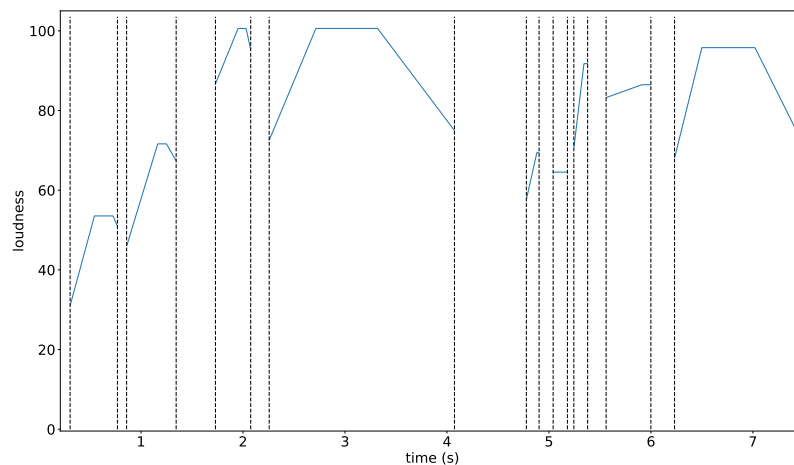
FIGURE 5.28: Example of generated loudness curve. Only the vowels segments, delimited by vertical bars, are shown for clarity

final loudness is applied on the synthesis using a time-varying gain computed to match this target loudness, as will be explained in the next chapter, section 6.2.3.

5.7 Evaluation

The evaluation of singing style modeling is a difficult task. A first reason, as already evoked, is that only a few of the many characteristics implied in the perception of a singing style are modeled, and one might thus wonder to which extent using only those features already allows to properly recognize a singing style, and to which extent the style is already contained in the score itself or is related to the singer's own interpretation or to personal timbral characteristics which are not modelled.

Another reason is that there is no really objective measure that would allow us to quantitatively assess the proper modeling of a singing style, which should thus be evaluated by means of subjective listening tests. A possible idea for an objective evaluation may be to use a K-folds cross-validation, by generating the expressive parameters for songs extracts not included in the learning data, and compare the values of the generated parameters to the real ones from the recording. But in case the values are different, this would not necessarily mean that the style is not properly modeled, as singing is not a deterministic process and nothing ensures that both interpretations might not be valid for this style.

In [STK10], the evaluation of the style's perception was only assessed for 1 style ("deep bendy" children songs). But in order to assess how well the proposed approach allows to capture and reproduce the characteristics of each style based on the modeled features, a better test would be to synthesize a same song in several styles and see if each style is well perceived and recognized among other ones in a listening test. But, as said previously, it is difficult to find a score that is well suited to be interpreted in different singing styles, as the score itself may already be oriented towards a certain style. The synthesis database should also ideally be well suited, in terms of timbre characteristics, for the different styles to be synthesized. These 2 conditions are hard to fulfil if the styles to be compared are too different

from one another. But on the other hand, the less different the styles, the harder it becomes for the listener to differentiate them in the synthesis.
For those reasons, one may probably not expect listeners to differentiate the styles modeled with a very high rate, based only on the prosodic features and using the exact same score, lyrics, and synthesis database for all styles.
However, besides the modeling of the singing styles, another goal of this work was also to improve the expressiveness of the synthesis, which may be easier to assess, as there is no need to compare the different styles for this purpose.
In order to evaluate those 2 aspects (the recognition of the singing style modeled and the improvement in expressivity with the addition of context-dependant variations), several listening tests have been conducted, which we present in the following sections. For this purpose, the proposed approach to style modeling has first been integrated into our *ISiS* synthesis system, and a style model has been built for each of the 4 styles in our corpus.

5.7.1 1st evaluation

5.7.1.1 Test design

For a first evaluation of the proposed approach, only the f_0 and phonemes' durations were modeled, and the style models were built from a single song from the corpus (other than "les feuilles mortes") for each style. The details of this first evaluation have been described in [ACR16b]. Note however that, although the approach remains the same, some improvements described in this chapter have been implemented after this publication (use of a different normalization, addition of a few contextual factors, use of weights on target parameters, and use of a distance function instead of a random selection on the leaves of the tree) and were thus not included in this first evaluation.

The first goal of this work being the modelization of singing styles, a first test aimed at measuring the recognition rate of the style modeled on synthesized singing. For this purpose, the chorus of the song “Les feuilles mortes” has been split into 4 parts (of around 15s each). For each part, the original interpretations in 2 styles were first presented (the 2 male or 2 female styles). Then, a synthesis produced using one of the 2 corresponding style models was presented, and the user had to guess which style was used for the synthesis, among the 2 possibilities, in an ABX testing procedure.

A second test was designed to assess the gain in expressivity when using the proposed approach to predict parameter values from contexts, compared to the use of a default configuration, where similar parameters are used for all contexts. The same song extracts that for the first test were used. For the default configuration, the parameters used for each f_0 segment were the mean values computed from the learning data of each style. For the vibrato, the sustain segments that don't carry any vibrato were not used to compute the mean parameters values. For the phonemes durations, the default durations computed on the synthesis database, as explained in the previous chapter, were used. Then, the listeners were asked to compare those 2 configurations (default values without context and style model with context), presented in random order, and rate their preferred interpretation on a 0-3 scale, based on the perceived expressivity (also defined in the instructions as

"liveliness", or "musicality") of the synthesis, using a standard CMOS procedure, similarly to the previous test presented in section 4.4.5.

The 2 tests have been conducted on 22 participants listening with either headphones or earphones through a web interface (similar to that shown in figure 4.20). All synthesis were generated using the SVP engine, and the 2 female and the 2 male styles of our corpus were used for both tests, using respectively the MS and RT databases. However, for the 1st test on style recognition, the female styles (Piaf and Greco) were not compared with the male styles (Distel and Le Roux), as the voices and pitch ranges are too different. Only 2 pairs of styles were thus assessed: LeRoux-Distel and Piaf-Greco. Furthermore, although the song is the same, their are some non-negligible differences in the interpreted scores between the 4 singers, which are also related to the style of each singer. Those differences concern the tempo, some rhythmical deviations and notes insertions. In order to compare the synthesis on a similar basis, and not influence listeners with stylistic differences related to the symbolic domain, which are not modeled in this work, we used for the 1st test an average score of the 2 singers for each pair. The sounds used in this evaluation can be found on the web page at the url[8], and the original listening test with the full instructions can be found at url[9] (note that in the web page, the test described here as 2nd was actually presented 1st to the listeners).

5.7.1.2 Results and discussion

The results of the first test are shown in table 5.3. The overall mean recognition rate is only 58.9% but gets up to 76.3% for the Piaf style, which suggests that the f_0 variations and/or phonemes' durations are characteristic features for this singing style, while those features may not be sufficient to differentiate well other styles. The percentages of good answers for the presented A-B pairs of styles are also given in the second row, and the p-value obtained for the Piaf-Greco pair indicates a significant result in regard to the random hypothesis. This result shows that we managed to capture some of the stylistic characteristics allowing to discriminate those 2 styles, while only modeling the f_0 and phonemes durations. We assume that this is mainly due to differences in the vibrato and transitions parameters (vibrato amplitude and frequency, transitions durations, and amplitude of preparations in upward transitions), which appear to be rather salient features for the Piaf and Greco styles. However, the results also suggest that those parameters alone are not sufficient to recognize a style very well, and thus encouraged us to pursue our research by including more features in our modelization, starting with the loudness.
The different original interpretations of the song presented for this test had important differences in rhythm and pitch range, which may also help to explain the low rating obtained for the LeRoux-Distel pair, as it was difficult for the listeners to focus only on the f_0 and phonemes durations, despite the fact that we used an average score for the synthesis to avoid favouring one style due to similarities in the score itself.

[8] http://recherche.ircam.fr/anasyn/ardaillon/IS2016/listTest/demo.php
[9] http://recherche.ircam.fr/anasyn/ardaillon/IS2016/listTest/

<table>
<tr><th>style</th><th>all</th><th>Le Roux</th><th>Distel</th><th>Piaf</th><th>Greco</th></tr>
<tr><td rowspan="2">recognition rate</td><td>58.9%</td><td>52.3%</td><td>55.9%</td><td>76.3%</td><td>54%</td></tr>
<tr><td rowspan="2"></td><td colspan="2">54.2%</td><td colspan="2">63.6%*</td></tr>
<tr><td>p-value</td><td colspan="2">0.365</td><td colspan="2">0.024*</td></tr>
</table>

TABLE 5.3: Singing style recognition rates for 1st test (*significant results)
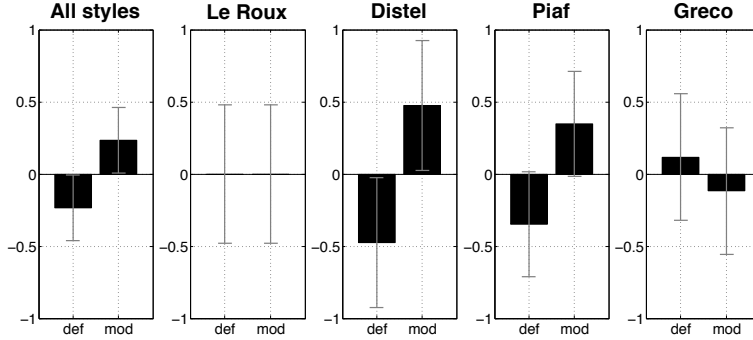
FIGURE 5.29: CMOS scores for default settings (def) vs. style models (mod) for the 2nd listening test of the 1st evaluation

Figure 5.29 shows the results of the second CMOS test on the assessment of expressivity, with confidence intervals of 95%. Although not very strong, a positive tendency in favour of the proposed model is observed for the global result. However, the results for each style differ a lot. Especially, the preference is quite clear for the Distel and Piaf styles, while no significant difference is observed for the Le Roux or Greco styles. A possible explanation for the fact that there is no preference for the Le Roux style is that his style is quite smooth and homogeneous such that the variations of the parameters with contexts remains limited and thus don't make a big difference with the use of default parameters. For the Greco style, a possible explanation is that it often presents quite unusual and exagerated downward inflexions (preparations) in transitions (such a shown in figure 5.23) that are not so much present in other singers, such that listeners may sometimes have preferred the default configuration in the synthesis for which those inflexions are limited. We may also expect the results to get better using longer test sounds, as the repetition of the same parameters may become more obvious for the default configuration. But longer sounds are harder to memorize to assess the differences, and we thus chose to keep them relatively short. The contexts present in the song used for each style may also not cover well all target contexts for the synthesis, and we may expect that the results improve by using more songs from the corpus to build the style models. However, the obtained results show that the proposed method can already, in some cases, improve the expressivity of the synthesis, compared to using a default configuration without additional input from the user. Considering that each singing style was built from a single song and that we only modeled the f_0 and phonemes durations, many other stylistic aspects being left apart in this study, the results obtained in these first tests are encouraging.

5.7.2 2nd evaluation

After integrating the various improvements described in this chapter and extending our style corpus to 3 songs per style, a 2nd similar evaluation has been conducted,

using the same styles and songs extracts. In this 2nd evaluation, all the songs from the corpus have been used to build the style models, except the chorus from "Les feuilles mortes", so that the extracts used in the test were not used in the learning stage. Note that similarly to the 1st evaluation, only the f_0 and phonemes' durations were modeled, the modeling of the intensity being judged not robust enough yet for a proper evaluation.

5.7.2.1 Test design

This 2nd evaluation also consisted in 2 tests to assess both the perception of the singing style and the expressivity in the synthesis.
The 1st test was however different from that of the 1st evaluation, using a CMOS procedure instead of ABX. For this purpose, an extract of an original recording of a target style was first presented. Then, for each extract, listeners were asked to compare a pair of synthesis of the same extract, and rate them according to which sound sounded the more similar to the target original recording, in terms of singing style. 3 different possible configurations were used to generate the parameters for the 2 synthesis in each pair: the use of the target style model corresponding to the presented original recording (labelled "target" or with the target name below the plots in figure 5.30); the use of default mean parameters from the target style without contextual dependency, similarly to the 1st evaluation described above (labelled "def"); and the use of the other non-target style model (labelled "other" or with the name of the non-target singer. e.g: If the "target" style is Piaf, the "other" style is thus Greco and vice-versa, and similarly for Le Roux and Distel). As this time a single target style was presented for each pair, there was no need to average the musical scores between different styles as was done in the previous evaluation. The scores used for the synthesis were thus built from the annotations of the corpus to match the notes in each target extract (in terms of midi notes and durations). An advantage of using a CMOS test to evaluate singing styles is that it allows to obtain a more graduated assessment than with the ABX procedure.

The 2nd test of this evaluation was similar to that of the 1st evaluation, aiming at assessing the perceived difference in expressivity when using the style models to predict the parameters from the contexts compared to the use of average default parameters.

This evaluation was conducted on 46 participants listening either with headphones or earphones. All synthesis for this evaluation were generated using the PaN engine with the MS and RT databases. In order to limit the duration of the evaluation, only 15 and 10 randomly-selected pairs were assessed in each test (from respectively 48 and 16 possible pairs) by each listener. The sounds used in this evaluation can be found on the web page at the url[10], and the original listening test with the full instructions can be found at url[11].

5.7.2.2 Results and discussion

Figure 5.30 and 5.31 below show the results of those 2 tests. As a main result, one can observe in figure 5.30 that the Piaf model is well recognized in the synthesis,

[10] http://recherche.ircam.fr/anasyn/ardaillon/singingStyles2017/demo.php

[11] http://recherche.ircam.fr/anasyn/ardaillon/singingStyles2017/index.php

as it is clearly perceived as closer to the target Piaf style than the synthesis using the Greco style and the default setting. The default setting is also rated as closer to the target Piaf style than the Greco model. For the target Greco style, the Greco model has also been better rated than the Piaf model, but no significant difference has been found with the default configuration. Those results are coherent with our 1st evaluation and confirm that we managed to model some of the stylistic features of the Piaf and Greco styles, although the use of contexts don't seem to make much difference for the Greco style. Unfortunately, no improvement can be observed for the Le Roux and Distel styles.
An hypothesis to explain this apparent limitations for the Distel and Le Roux styles is that the intensity variations are of particular importance in the characterization of those 2 styles, but has not been modeled in this test. Moreover, besides the fact that the intensity variations are not modeled, one may assume that this also have implications on the perception of the f_0 variations. It seems indeed that fluctuations like attacks or vibrato are perceived differently depending on the intensity, such that they may be properly modeled but not considered as such because of a higher loudness level or a lack of intensity variations compared to the original recording. Additionally, Some remaining artifacts in the synthesis may also sometimes impact the proper perception of the f_0 and durations parameters et degrade the results (e.g. when important downward transpositions are required during attacks or transitions).
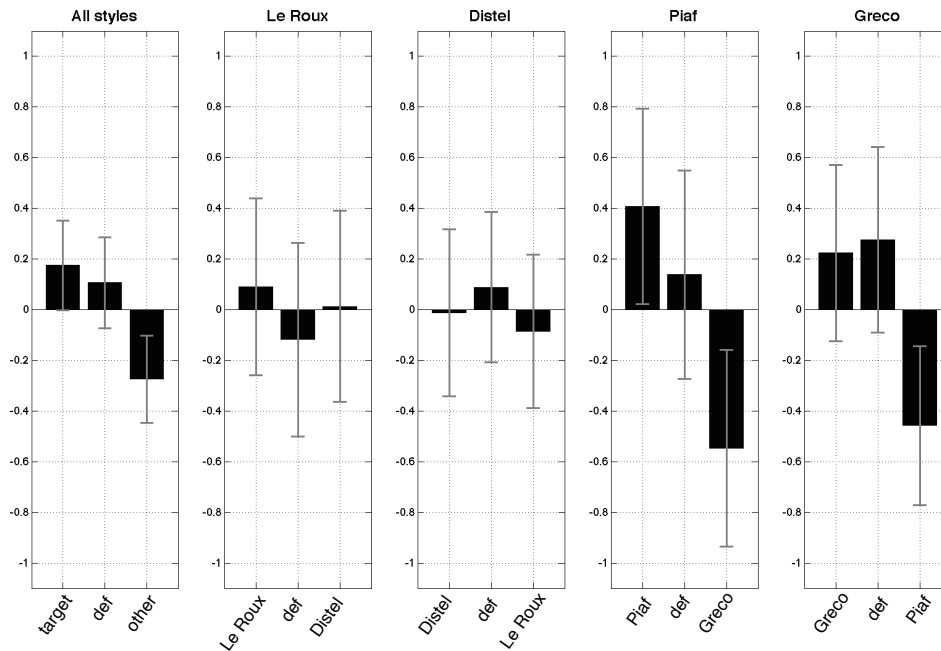
FIGURE 5.30: CMOS scores related to the perception of the target singing style in synthesis for the 1st listening test of the 2nd evaluation. "target" stand for target style model, "def" stands for default setting using averaged values from target model, and "other" stands for the "non-target" style model (e.g. other is Greco if target is Piaf)

Regarding the 2nd test, the results seem to have been well improved for the Piaf style compared to the 1st evaluation, which suggests that the various improvements and the use of a bigger corpus have been effective to model context-dependant

variations for this style. However, no improvement are observed for the other styles, especially for the Distel style for which the results have been surprisingly degraded. A possible explanation is that in this test the scores were extracted from the original recordings, and thus had a lower pitch and slower tempo for the Distel style. Due to the resulting important durations of the sounds, it may thus have been more difficult for listeners to evaluate the extracts in their globality, and they may have focused more on localized artifacts. The use of a slower tempo might also have highlighted more the lack of dynamic variations on long sustained notes.

Note that the results obtained in these evaluations (especially for the Distel style) highlights the difficulty of conducting and interpreting such evaluations, as it is difficult to assess whether the differences between the 2 evaluations come from limitations of the proposed method or from some bias of the evaluation procedure itself.
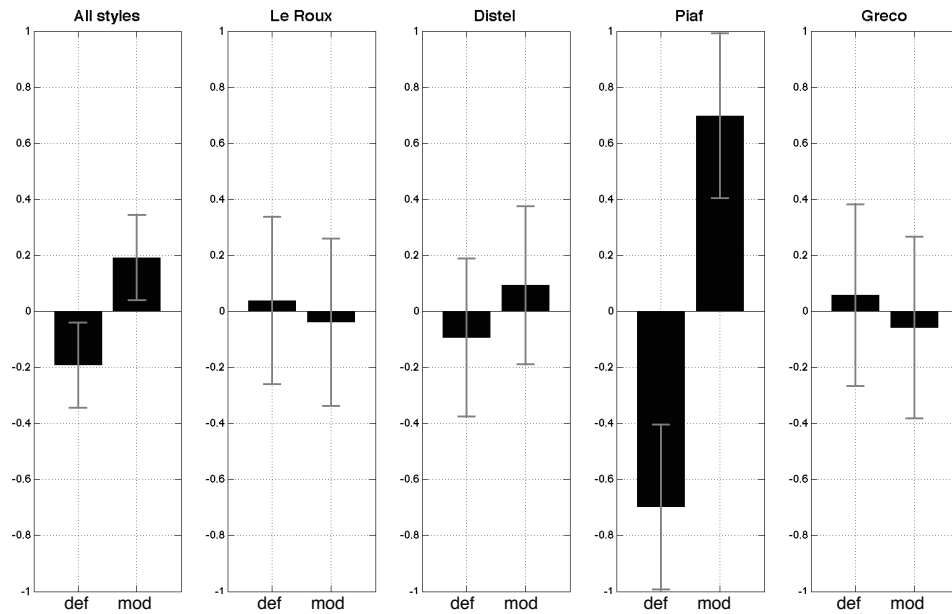
FIGURE 5.31: CMOS scores related to perceived expressivity for default settings using average parameters of target style (labelled "def") vs. style models (labelled "mod") for the 2^{nd} listening test of the 2^{nd} evaluation

5.8 Summary and perspectives

In this chapter, we first discussed the notion of singing style and the many aspects involved in its perception. We saw that this notion is hard to precisely define as it is somewhat subjective and can be perceived at different levels, with a "generic model" representing a broad stylistic category, and a "stylistic model" representing the style of a specific artist inside those generic categories. From these considerations, we decided to follow an "inductive" approach, from the example to the generic model (similarly to [Cha13]), in order to have a maximum of consistency in the various aspects of the styles to be modeled. Based on this approach, we constituted a corpus with recordings of 4 famous French singers representative of different stylistic categories. Among the many aspects involved (symbolic,

prosodic, timbral, ...), we focused in this chapter on the prosodic aspect, encompassing the f_0 and loudness fluctuations as well as phonemes' durations. We then proposed a new approach for automatically generating expressive interpretations from a score and lyrics, and model singing styles, based on parametric templates selection, that has been the subject of 2 publications [ACR16a; ACR16b]. In this approach, the f_0 and loudness parameters and phonemes durations are extracted from annotated recordings, along with a rich description of contextual informations, and stored to form a stylistic database of parametric templates. In order to take into account the potential variable importance of contextual factors in the interpretative choices of singers of different styles, this database is then used to build styles models using decision tree-based context clustering. At the synthesis stage, appropriate parameters are then selected according to the target contexts obtained from the score, by picking templates from the corresponding leaf of the trees. The proposed approach has been finally evaluated in listening tests to assess both the gain in expressivity compared to the use of default parameters without manual tuning, and the ability of listeners to recognize the styles modeled in the synthesis. The results of this evaluation showed that the proposed approach can, in some cases, improve the expressivity of the synthesis, and that we managed to capture some of the stylistic characteristics allowing to recognize certain singing styles in the synthesis. However, those results remain limited and vary a lot depending on the target style. In future works, a priority would be to improve the modeling of the intensity, which seems of particular importance for the modeling of the Distel and Le Roux styles.

The proposed approach aims at combining the advantages of the various state-of-the-art approaches while avoiding some of their drawbacks.
Similarly to HMM-based approaches, the use of parametric representations with decision tree-based contexts clustering allows to benefit from a rich contexts description that can hardly be used in units selection-based approach which rely on costs functions. But the use of specific templates of f_0 or intensity segments, extracted from recordings, without short-time statistical modelisation avoids the oversmoothing problems of HMM-based approaches by using variations close from the real original curves as in unit selection-based approaches.
As said previously, a particular advantage of our approach over HMM or unit selection-based systems is also that the parametrization of the f_0 and loudness curves allows the user to intuitively refine the result obtained with the automatic approach. In cases where the result of the automatic procedure is not found to be optimal, it nevertheless already allows to alleviate the need of fastidious manual tuning to obtain a satisfying result. With this approach, the final result is also not constrained by the material from our corpus, as the parameters can be adjusted to generate expressions that are not present in the corpus.
Finally, our models being parametric, one can also easily constrain the result based on specific knowledge, as in rule-based approaches, which we do for instance regarding the placement of f_0 transitions according to the phonemes' positions.
Note that the use of a parametric model to represent the expression contours has been mentioned in [Umb15] as a perspective for improvements over the proposed unit selection-based approach, which tends to confirm the relevance of our approach (although we started working on our parametric approach before the publication of this thesis).
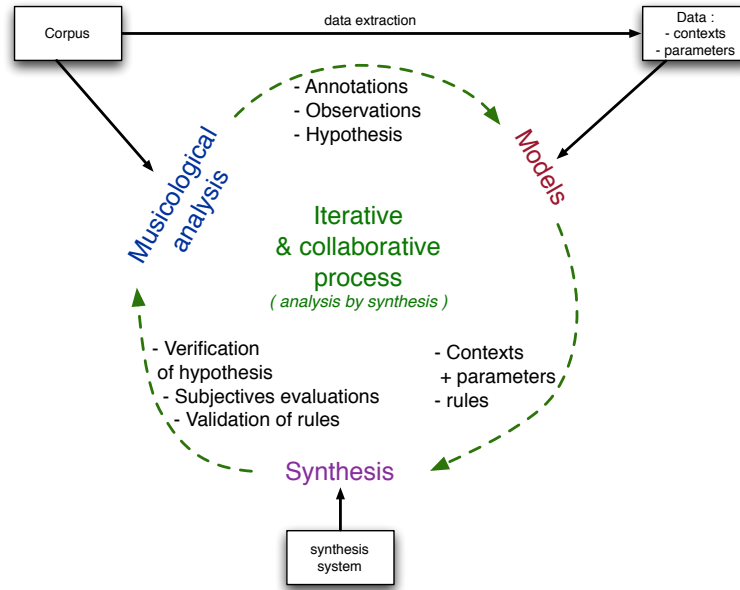
FIGURE 5.32: Illustration of the collaborative process between musicology and singing voice synthesis

Furthermore, an advantage of using decision trees in our approach is that, contrary to other machine learning techniques like neural networks, they can be used with fewer data and are easily readable and interpretable by a human. Beyond their use for synthesis, the reading of the trees built from the data might thus also be useful for musicological purposes, for instance to verify some hypothesis about the importance of the various contextual factors in the interpretative choices of singers. Using a large set of contexts, the most influential ones are automatically found, regarding a specific expressive feature to be studied, and the sound extracts from the corpus matching those contexts can be automatically retrieved, which can ease the systematic and large-scale study of a musical corpus for musicologists. Furthermore, using an analysis-by-synthesis approach, hypothesis can be formulated from the observation of recordings, which can be verified using synthesis, and the models can be further refined based on the results. This way, both fields of singing voice synthesis and musicology can benefit to each other, in an incremental process, as illustrated in figure 5.32.

Besides learning the style of an existing singer, an interesting perspective for the proposed approach would be to create a new singing style from scratch, only based on the inputs of the user, without using any recordings. For this purpose, one might start with a default configuration. Then, while synthesizing some songs, the user may manually adjust the parameters. Based on the user's choices, the system could then start building a new custom style model for this user, iteratively updating the decision trees each time the user modifies some parameters. This way, the system could then propose more and more appropriate parameters to the user, based on his past choices (assuming the user is rather consistent in his interpretative choices).

However, the main limit of our approach is that the shapes that may be produced by our f_0 and intensity models are limited by the restricted set of parameters used. Although those models fit rather well the variations encountered
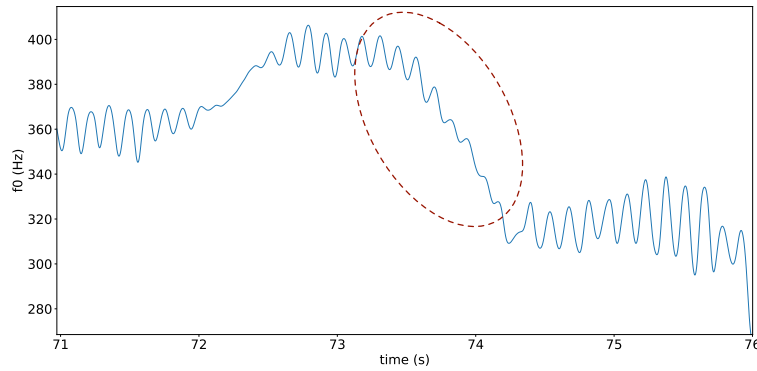
FIGURE 5.33: f_0 extract showing a downward transition carrying vibrato in a recording of Edith Piaf
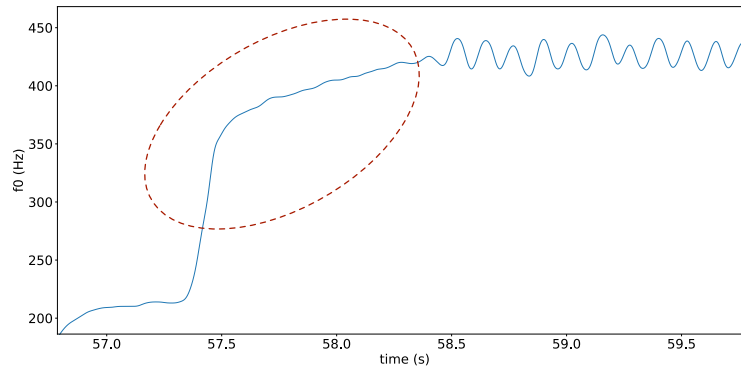
FIGURE 5.34: f_0 extract showing a "broken" 2-steps transition with a "knee" on the right part in Piaf

in recordings in most cases, each singer may use specific types of variations that are never or rarely encountered in other singers and that the model may not be able to fit well. This could be seen as some kind of oversmoothing, as for instance 2 transitions with different shapes but with similar characteristics (e.g. durations) may be parametrized similarly for the model and thus reproduced with a single shape. An example is the ending of some long notes followed by a note with a lower pitch in Piaf, in which case the f_0 curve sometimes falls down towards the next note's pitch while the vibrato is still present, as shown in figure 5.33. In our model, vibrato is only present on sustains segments, while the fall of the f_0 would rather be seen as a downward transition. To represent such a case, we would thus need to enable the vibrato to continue during the transition, which is currently not possible in our model.

Another example in Piaf is some upward transitions presenting a knee followed by a rather flat ascending slope on the right part, as shown in figure 5.34, which can also not be accurately represented with the provided parameters in our f_0 model.

Representing such cases with our approach would require to add new parameters to allow a higher flexibility of the generated curve's shape. But this would also increase the complexity of the model, which would be more fastidious to use for manually tuning the parameters, and would also make the automatic learning of singing styles more difficult. Moreover, one may always expect to encounter new specific cases when studying new singers which would require to adapt the model. But this limitation only concerns a few specific and rather rare cases, and

we assume that our model is able to reproduce most of the important features and typical variations found in Western-European musical styles.
Another related limitation of our approach is that it is quite dependant on the accuracy of the parameters estimation, and thus on the corpus annotation, and especially the consistency of the f_0 segmentation. In some particular cases, the better segmentation strategy is not straightforward, like for instance in the case represented in figure 5.34, where the flat ascending part may be either considered as the end of the transition segment, or the beginning of the sustain segment. In such case, the chosen segmentation would result in different parameters for both segments. Due to possible parameters estimations errors, some outliers with inappropriate parameters values may also be selected and sound unnatural in the synthesis. Such outliers should thus ideally be discarded beforehand.

To circumvent such problems, an interesting perspective would be to somehow combine our parametric approach with a unit selection approach, by parametrizing real contours and learn the contextual dependencies using decision trees similarly to our current approach, but using the real contours for synthesizing the final curve instead of using our B-splines-based model. This way, the approach would keep the advantages of using decision trees (adaptability and use of a rich contexts description) rather than using a cost function, but would benefit from the precision of the real contours.
In order for the user to keep some control and be able to modify the result, the target parameters could also be modified intuitively like in our approach, and the curve may then be modified to approximate those new target parameters by directly streching or warping the original f_0 curve. In order to avoid too important transformations, another contour that better matches the new target parameters could also be selected from the database to replace the original one before being further transformed.

As said in section 5.4.3, some contextual factors have not currently been exploited due to current limitations of our system and a lack of data or specific annotations. Especially, semantic informations like the grammatical category of a word or the position of a syllable in a word, as well as rhythmical informations like the position of a note in a bar may be of particular importance, as some notes may be particularly emphasized if they are at the beginning of an important word, or on a downbeat. Thanks to the help of our musicologist collaborator, some semantic annotations have now been added to the corpus, but have not been exploited yet. In future works, those annotations may thus be exploited and hopefully further improve the results of our approach.

In our approach, the phonemes durations, f_0, and loudness are modeled rather independently from each other. But in real singing, they appear to be often correlated. For instance, a note may be accentuated by simultaneously having a high loudness level, as well as a long consonant and a big preparation's amplitude before the vowel's onset. Vibrato also tends to be wider for high loudness levels [BS02]. In a first attempt to take this inter-correlation of parameters into account, we already used the consonants identities and durations as contextual factor for predicting the f_0 transitions' parameters. Similarly, one may also first predict the loudness so that the loudness level of a note could be used as a contextual factor to determine the consonants' durations and f_0 parameters. Another possibility might be to tie the loudness and the vibrato parameters together, for instance by first predicting the

loudness parameters and then choose the vibrato parameters corresponding to the same note in the corpus than that of the chosen loudness parameters.
The "intra-correlation" of each parameter could also be better taken into account. For instance, the amplitude of overshoot in upward transitions are often related to the presence and amplitude of vibrato in the following sustain segment. Currently, the f_0 parameters are chosen for each segment independently of the surrounding ones. To better take into account this possible inter-correlation between f_0 segments, one may for instance first predict the transitions durations and vibrato parameters in a first step, and then predict the amplitudes of preparations and overshoots only in a second step, using the parameters of the surrounding segments (e.g. vibrato amplitude) as contextual factors.
With a similar idea of a 2 steps procedure, one could also first predict only the global vibrato amplitude, and then predict the other vibrato parameters in a second step, according to the predicted amplitude, as is done for loudness, which would have the advantage of using less target parameters for building the decision trees.

Finally, another perspective would be to use a more refined distance measure for selecting the templates on the leaves of the trees, based on the cost functions defined in [Umb15], thus introducing some more knowledge in the selection process while still benefiting from the learning capabilities of the decision-tree-based contexts clustering. Another possible solution, rather than using multi-target regression trees, might also be to first cluster the segments into a set of prototypic classes (e.g. very short transitions, long transitions with big preparation, ...), then use a simple classification tree to predict the class to use, and finally use a cost function to choose the best example in the prototypic class according to the contexts.

Regarding the loudness, we proposed a first approach to model both the note-to-note variations along a musical phrase and the intra-notes dynamic variations. However, this approach has not yet been thoroughly tested. From preliminary informal tests, we found out that the proposed approach often improves the expressiveness of the synthesized voice by introducing dynamic variations. However, the generated variations of the loudness level are still sometimes too important, which sounds somewhat unnatural in some cases, and the proposed approach should thus be further improved to obtain better results. It remains however unclear where the limitations come from, as it may also be partly related to the accuracy of the loudness analyzed on polyphonic music, which is quite approximative. It is also not easy to properly evaluate the control of the loudness without a realistic intensity transformation, as although the applied variations may be appropriate, the transformation might be judged too unnatural for its control to be considered as correct (e.g. if a simple change of gain without timbre modification is applied).
However, a possibly better approach regarding the control may be to use a multi-layer model for the loudness, by first generating a global loudness contour over the course of a whole musical phrase, and then add the more local intra-notes variations on top of this curve.

But using only the prosodic parameters is also not sufficient to represent all the stylistic characteristics of singers. In future works, the symbolic and timbral aspects should also be modeled for a more complete representation of the singing styles.
Regarding the symbolic aspect, the authors in [Umb+15] suggested to use rule-based techniques *"as a preprocessing step to modify the nominal target score so*

that it contains variations such as ornamentations and timing changes related to the target style or emotion".

In the next chapter, we will present some works on several types of timbre transformations in order to further improve the expressiveness and naturalness of the synthesis, and modify the vocal quality, which is necessary to represent more varied singing styles.

Chapter 6

Expressive timbre transformations

6.1 Introduction

In previous chapters, we first saw how we can generate the voice signal using concatenative synthesis, and then how to generate appropriate control parameters to produce a more expressive synthesis. Although those 2 aspects already allow synthesizing singing voices with a rather good quality, they still lack of naturalness and expressiveness compared to real singers, as has been assessed in [Feu+16].
An appropriate control is indeed not sufficient to obtain a truly natural result if the timbre is not coherent with the control parameters, especially regarding the pitch and intensity. For high-quality synthesis, it is therefore necessary to provide means to generate appropriate changes in timbre that are coherent with those control inputs. Past studies have already permitted to gather knowledge about how various spectral features of voice may change during singing, which could be exploited in synthesis systems by defining a set of rules, as suggested for instance in [HSW11] regarding formant tuning for pitch transformations. But if such rules are easy to implement in formant synthesizers like [Feu+17], where all source and formants' parameters can be explicitly controlled, this task is much more complicated for systems based on signal transformations like our system.
In the previous chapter, we also evoked many features that may be used by singers as vectors of expressivity related to specific singing styles or singers' identities, some of which are related to the voice timbre. Some examples are the use of different registers or specific techniques (e.g. belting, fry, ...), vocal roughness, breathiness, ...

In the present chapter, we focus on the modeling of those 2 aspects of timbre, both for generating a more natural synthesis, and to extend the palette of possible expressive vocal effects necessary to synthesize a wider variety of singing styles, which are major current challenges to improve state-of-the-art SVS systems. The contributions presented in the following thus concern the development of rules and algorithms to produce such expressive transformations, that may be later integrated into our synthesis system to improve its naturalness and extend its expressive capabilities. In a first section, we will focus on the problem of producing realistic intensity transformations, considering various physiological aspects involved. Then, some investigations on 2 new approaches to model vocal roughness will be presented.

6.2 Intensity transformation

Regarding the timbral features related to intensity, various aspects have to be considered. Voice intensity is highly related to the notion of vocal effort, and as pointed out in several sources [PD16; TE00; LB13], a vocal effort increase is physiologically linked to an increase in subglottal pressure, an increase in vocal-fold tension, and a wider mouth opening. From the view point of the source-filter model, the well-known effect of increasing the vocal folds' tension on the source's spectrum is an increase of the glottal formant frequency and a decrease of the spectral tilt, while the main effect related to a wider mouth opening is an increase of the 1st formant's frequency (F_1), as observed in many studies [PD16; TE00; LB13; LD99b; Hub+99; Sun90]. Additional aspects to be considered may be the level of aspiration noise, or the singer's formant's prominency [Sun01]. For changing the intensity of a voice, it is thus not sufficient to only modify the sound level. For our purpose, 2 different types of approaches to produce these effects have been considered.

Spectral morphing approach:

A possible approach to create transformations of vocal intensity is spectral morphing, that uses target templates recorded at different levels of low and high vocal efforts, as proposed in [SG03; Tur+05; DSC13] for diphone speech synthesis. We explored the potential of such a morphing approach to be used in our singing synthesizer, as presented in [DAR16a]. Note that this approach may also be used for pitch transformations, as the spectral envelope is also dependant on the pitch. The suitability of this approach mainly relies on a proper estimation of the spectral envelope for its integration into a singing synthesizer (which is especially a problem for high-pitched female voices, as already discussed in section 2.2.3). Indeed, if the target envelope is not properly estimated, this may lead to an unnatural result when this envelope is applied to a voice signal with a different pitch. In order to alleviate this problem, we proposed to use a multi-frame analysis (MFA) of the spectral envelope, by combining spectral information from several successive frames. For singing voice, this approach allows to exploit the pitch changes related to vibrato, which sweeps through the spectral envelope across time. Assuming that the VTF is rather stable across a vibrato cycle, this can lead to more accurate estimates of the spectral envelope than the traditional single-frame approaches presented in section 2.2.3.

Based on this idea, 2 variants have been tested, mainly aiming at simplifying previous over-complex approaches to such multi-frame analysis. The 1st method, named SDCE-MFA (for *Simplified Discrete Cepstral Envelope for Multi-Frame Analysis*), consists in a mathematical simplification of a previous approach presented in [SK03b], which is an MFA version of the least-square (LS) cepstral solution [GR91; CCM01] mentioned in section 2.2.3.1. The 2nd method, named Linear-MFA-Lift, consists in low-pass liftering the envelope obtained by linear interpolation of the harmonic peaks of the pre-aligned frames, which is computationally lighter than the SDCE-MFA method.
The analysis procedures for those 2 variants have been detailed in [DAR16a] and [DAR16b] and the morphing approach using those multi-frame analysis has been integrated into our ISiS system for testing this approach in the context of singing synthesis. However, although I contributed to the publications of these methods

and to the integration of the algorithm into ISiS, this study essentially results from the work of Dr. Gilles Degottex, who contributed to the ChaNTeR project as a post-doctoral researcher, and thus can't be accounted as a contribution from this thesis work, and won't be further detailed here.

Although the stability assumption of the spectral envelope throughout a vibrato cycle may not be fully satisfied, this approach nevertheless gave encouraging results in listening tests. In [DAR16a], spectral morphing has been evaluated on sustained vowels both for pitch and intensity transformations, using the envelopes obtained with both MFA approaches, and the results showed clear improvements for both approaches over the use of the True-Envelope algorithm, thus supporting the use of multi-frame analysis methods for spectral morphing-based transformations.

A particular advantage of the spectral morphing-based approach is that it includes both the effects related to the glottal source and to the vocal tract, without requiring advanced knowledge of the underlying physiological mechanisms or signals properties.

However, the target envelopes used for morphing are taken from stable parts of vowels only and thus don't allow to correctly reproduce the timbre variations occurring on coarticulation parts where the envelope is less stable. In our implementation, we used a linear interpolation between the original and target envelopes around the stable parts to avoid timbre discontinuities, but this tends to create unnatural transitions at vowels' boundaries, when used in the real context of singing synthesis. An example of such morphing-based transformation is given in sound 6.1. A solution to this problem may be to use aligned parallel recordings at various intensities for the full database. But this would be much heavier compared to using recordings of vowels only. Moreover, recordings at different vocal intensities are not always available, and it would thus be interesting to be able to apply intensity transformations on any voice, without the need of such recordings. Another limit of morphing-based approaches for intensity transformations is that it doesn't allow to generate the missing higher harmonics that are accompanying the decrease of the spectral slope of the source for weak-to-loud transformations, and the application of the target spectral envelope on the whitened original signal may thus amplify the noise in high frequencies instead of sinusoidal components, thus increasing the hoarseness of the voice.

Parametric approach:

To overcome the limitations of morphing-based approaches, a possible alternative is to use a more knowledge-based parametric (or spectral modeling) approach to transform the sound without the requirement of additional recordings, by decomposing the global effect of intensity variations into several physiologically meaningful independent components related to the modification of the source, the VTF, the noise level, etc..., that can then be adapted to the context of vowel, gender, or singing style.
In that direction, past studies investigating voice quality modifications such as [AD03] and [ADC98] have primarily focused on the source-related spectral characteristics, while only few recent studies based on parametric approaches like [PD16] and [Mol+14] started considering the effect of the vocal tract for intensity

transformations.

In our work, we investigated rule-based approaches to modify both the voiced source spectrum and the VTF, that we present below.

6.2.1 Glottal source transformation[1]

As said previously, two effects have to be considered for the modification of the glottal source with intensity: the change of spectral tilt, and a modification of the glottal formant. A particular effect when decreasing the spectral tilt (for weak-to-loud transformations) is the emergence of new harmonics that rise above the noise floor in the high frequency range.

In some of the previously mentioned studies, authors approximated these behaviours using filters and non-linear time-domain transformations. However, using the PaN parametric synthesis engine introduced in section 3.5.2, we have access to the R_d shape parameter of the glottal pulse, which controls both the spectral tilt and glottal formant behaviours, and may thus be used for intensity transformations. When lowering the R_d value, the spectral tilt decreases (generating new high-frequency harmonics) and the glottal formant's frequency increases, as was already illustrated in figure 2.6. The use of the R_d parameter has already been investigated for breathiness and tenseness modifications (which are also related to intensity) in the context of parametric speech synthesis [DRR11b; HR15]. This relation between R_d and intensity variations is also mentioned in [Roe+12] in the context of singing voice.

In [Fan97], the author stated that the *"overall voice intensity [is] manifested by increasing E_e, and usually decreasing R_d and R_a"* (with $R_a = t_a/T_0$, following notations from section 2.2.2), and further suggested a *"general rule of covariation of 1dB in $1/R_d$ with 2dB in E_e [that has been found] to be typical of dynamic variations [...] as a consequence of varying voice effort"*. This rule is assumed to maintain a certain coherence in the evolution of the source's spectrum and its energy, and we thus chose to start by implementing this rule into our system, using the PaN engine, while limiting the R_d value to its usual range [0.3; 2.7], as described in [Dic16]. Note that such modification of the R_d parameter could potentially be also applied in the SVP engine by modifying the spectral envelope to reflect the change in the source spectrum (which is implemented as an option in superVP). This approach could thus modify the spectral tilt and glottal formant, but this would not allow generating the missing high-frequency harmonics for weak-to-loud transformations. The advantage of using PaN for this purpose is thus that those harmonics are also automatically generated by the LF source model when lowering the R_d value.

According to the results of the previously mentioned studies, the relevance of varying the R_d parameter with vocal intensity is not to be attested anymore. However, the remaining question is to which degree this rule should be applied, for each phoneme, according to a given target intensity level.
A first idea to answer this question was to measure the R_d values on vowels sung at

[1] The work presented in this section on glottal source modification was conducted with Maxime Dickerson in the framework of his masters' internship [Dic16]

different intensities, available in our databases. However, although the tendency is towards a decrease of R_d for highest intensity levels, the measured values did not show enough coherence to infer a clear correlation between R_d and intensity, and no generic behaviour could thus be hypothesized from those measures. Reasons for this may be that some errors may remain in the R_d analysis, or that the voice quality is not consistent enough between the different recordings. One may expect that a clearer correlation between the measured R_d and intensity could be obtained by averaging the analysis from many recordings. However, we currently don't have enough data on hand to verify this.

Our next idea was thus to rather measure the loudness variations induced by the change in the source parameters, and then invert the result to deduce the necessary increment in R_d and E_e to match a given target level for a specific vowel and f_0. This was done by synthesizing a set of source signals based on a grid of f_0 and $[E_e; R_d]$ values (incremented according to Fant's rule evoked above), and further filter them by a VTF corresponding to each vowel. Then, the loudness level was measured on each synthesized signal, and some regressions were used to deduce to which degree the rule should be applied for each vowel, according to a given f_0 and target loudness value, as detailed in [Dic16]. A preliminary subjective listening test was then conducted to evaluate the gain in naturalness obtained for intensity transformations using this approach. For this test, we generated crescendi on 3 vowels (/a/, /i/, /u/) using the inferred rule to match the loudness profiles of real crescendi. Then, listeners were asked to compare the result, in terms of naturalness, to the original crescendi recordings and to another synthesis for which a simple gain was applied to match the target loudness (as will be explained in section 6.2.3) without modifying R_d. The sounds used for this test can be found on the web page at url[2]. However, although informal listening suggested that the proposed approach should give more realistic results, the answers to this test confirmed this improvement only for the vowel /i/. Several hypothesis may explain this limited result:

- A 1st hypothesis is that the test was designed in such a way that listeners unfortunately didn't really answer the question asked. It appears to us, after analyzing the answers and feedbacks from listeners, that some of them may have rather assessed the aesthetic qualities of the voice, rather than the naturalness or coherence between timbre and intensity. It seems that modal voices using the original R_d contours (without applying the rule) may have been preferred to more tensed voices with more high-frequency content, characteristic of high loudness levels when lowering the R_d value.

- A second hypothesis is that the R_d parameter itself is not sufficient to fully represent the appropriate change in source spectrum, and some source shape parameters like t_a may need to be tuned more finely, e.g. to limit the amplitudes of the highest harmonics that may confer a buzzy quality to the sound.

- Another potential issue is that in this approach we used a fixed VTF, assuming that the loudness variations were primarily related to the source component. But the VTF should also vary with intensity (e.g. due to mouth opening), which would then also have impact on the loudness variations. This aspect may thus change the values obtained to control the degree of the

[2] http://recherche.ircam.fr/anasyn/ardaillon/testIntensitySrc2016/demo.php

rule to be applied for each vowel, which has not been considered here. For some vowels, the modification of the source to be applied may thus be over-estimated and judged as unnatural, because of too extreme R_d values. The fact that better results were obtained for the closed vowel /i/, for which mouth opening (and thus potential VTF variations) are more limited than for open vowels like /a/ tend to confirm this hypothesis. A more fastidious, but possibly better approach may thus be to manually tune the degree of the rule for each vowel and each voice according to the perceived timbre coherence, in an analysis-by-synthesis approach, rather than using a systematic approach like we did.

- Finally, as the change in the VTF has not been modeled in this experiment, one may assume that the lack of coherence between the source spectrum and the VTF could be judged as unnatural and thus doesn't allow to properly assess the result of the rule, although the transformed source may be closer to what would be expected for the target intensity level.

These results and hypothesis thus encouraged us to work on the modeling of the VTF variations related to intensity, in order to create a more complete effect, including both the contributions of the source and vocal tract, which we expect to be easier to assess and give better results.

Besides the voiced component of the glottal source, the noise level is also likely to vary with vocal effort and should probably be amplified when the voice gets louder, limiting the potential "buzzyness" that may arise due the rise of high-frequency partials. In the PaN engine, the noise level can easily be modified by simply applying a factor on the unvoiced component. In our previous tests, we simply used the variation of E_e as a gain for the noise component. However, further research on this aspect would be necessary to properly tune the noise level according to the overall intensity.

6.2.2 Mouth opening transformation

In order to sing louder, singers tend to open their mouth more widely, which changes the vocal tract's shape (VTS) and resonances, thus affecting the VTF. In this section, we show, by means of signals analysis and simulations, that the main effect of mouth opening is an increase of the 1^{st} formant's frequency (F_1) and a decrease of its bandwidth (BW_1). From these observations, we then propose a rule for producing a mouth opening effect, by modifying F_1 and BW_1, and an approach to apply this effect on real voice sounds. This approach is based on poles modification, by changing the AR coefficients of an estimated all-pole model of the spectral envelope. Finally, listening tests have been conducted to evaluate the effectiveness of the proposed effect.

6.2.2.1 Real signals analysis

As a starting point for this study, we analyzed spectral envelopes on the 15 French vowels from our RT database, sustained on 1 pitch (135Hz) and 5 levels of intensity, from pianissimo (*pp*) to fortissimo (*ff*). In order to observe the change in the vocal tract's resonances, we employed the following procedure for estimating the VTF:

- First, the DAP algorithm [EM91] with order 50 was used to estimate an all-pole model of the spectral envelope;
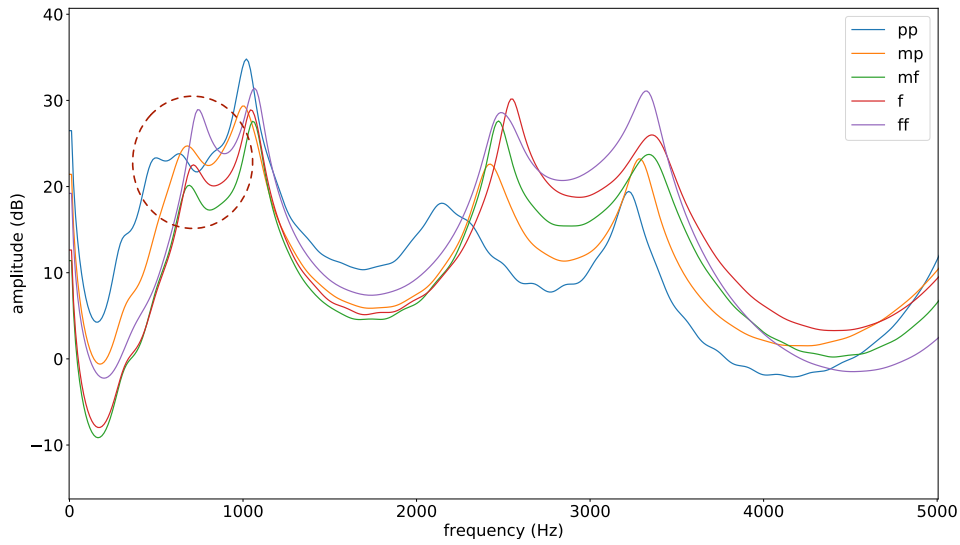
FIGURE 6.1: measured VTF for the vowel /a/ sung by RT on 5 intensity levels

- Then, the R_d parameter of the LF source model was estimated based on algorithm described in [DRR11a; HR14], and the contribution of the source was removed by spectral division of the DAP envelope with the spectral shape associated with the estimated R_d value.

The DAP algorithm is used here instead of the True-envelope as it better allows to differentiate the resonances of the vocal tract associated to the poles of the model. Informal observations of these analysis confirmed an increase of F_1 with intensity in many cases, especially for open vowels like /a/, and sometimes a decrease of BW_1. Figure 6.1 shows an example of such analysis for the vowel /a/. However, these observations present an important variability, depending on the vowel, but also maybe related to some limitations of the analysis algorithms, or to the consistency of the recordings across the various intensity levels. The same procedure was also applied on recordings of varying degrees of mouth opening (without intentional changes of intensity levels), but resulted in similar observations. Similarly to the source component, this variability makes it difficult to infer a precise rule to be used in a synthesis system, based on those observations only. For this purpose, we thus employed a simulation approach, as described in the next section.

6.2.2.2 Simulations

As seen previously, it is rather clear that F_1 should increase with mouth opening. But it is not clear to what extent, and most studies don't evoke a possible change of BW_1. Another possible approach to make assumptions on the voice characteristics behaviours is to use simulations.

As explained in [Wak73; MG76], there is a direct equivalence between the simple acoustic tube model of the vocal tract and linear prediction. It has indeed been demonstrated in [AH71] that *"a transfer function with P poles is always realizable as the transfer function of an acoustic tube consisting of P cylindrical sections of equal length"*. We can thus use this relation to estimate a Vocal Tract Shape (VTS) from an all-pole model, modify this VTS to simulate mouth opening, and convert it back to an all-pole model to observe the effect on formants

parameters (represented by the poles of the model).

From the acoustic point of view, reflection coefficients μ_k between the sections k and $k-1$ of an acoustic tube model are defined as:

$$\mu_k = \frac{A_{k-1} - A_k}{A_{k-1} + A_k} \tag{6.1}$$

where A_k and A_{k-1} are the areas of the k^{th} and $(k-1)^{th}$ sections of the tube.
As explained in section 2.2.3.2, an all-pole model can be defined by its transfer function:

$$H(z) = \frac{G}{1 + \sum_{i=1}^{P} a_i z^{-i}} \tag{6.2}$$

where the a_i are the AR coefficients of the model, P is the model order, and G is a fixed gain coefficient. And as explained in [MG76], the P filter coefficients a_i can be computed from the reflection coefficients μ_k of the equivalent P-sections acoustic tube model using the step-up procedure, which can be summarized by the following equation:

$$a_{ki} = \begin{cases} a_{k-1,i} & i = 0 \\ a_{k-1,i} + \mu_k a_{k-1,k-i} & i = 1, 2, ..., k-1 \\ \mu_k & i = k \end{cases} \tag{6.3}$$

for $k = 1, 2, ..., P$ with $a_{00} = 1$, where the $a_k i$ are the coefficients of the k^{th} order model, and P the final order of the model.
Conversely, the step-down inverse procedure can also be used to retrieve the reflection coefficients, and thus the area ratios describing the VTS by inverting equation 6.1, from the filter coefficients, according to the following equation:

$$\begin{array}{l} a_{k-1,i} = \frac{a_{ki} - \mu_k a_{k,k-i}}{1-\mu_k^2} \\ \mu_k = a_{kk} \end{array} \tag{6.4}$$

for $k = P, P-1, ..., 1$, $i = 0, 1, ..., k-1$, and $|\mu_k| < 1$. Thus, a unique discrete tube shape can be reconstructed from a transfer function polynomial of given order. Assuming that the source component has been properly removed, a reasonable estimate of the VTS can thus be obtained from the all-pole coefficients.
More details on those procedures and the relations between the acoustic tube model and all-pole model of voice are given in [MG76].

Simulation procedure:
In the following experiment, we used the open-source sparkNG software[3] [Kaw16], that implements those procedures in the matlab environment to convert back and forth between the VTS and its equivalent all-pole model, while providing a convenient GUI to manipulate, display, and store both. Figure 6.2 shows the interface of the software. In the top left panel, the VTS is displayed from the lips to the glottis as relative areas of the acoustic tube model; the top right panel shows the corresponding all-pole spectral envelope model. The frequency and bandwidth of each formant can be displayed just below the plot. The bottom left panel allows to draw a modification curve to be added to the original VTS, from which the all-pole model is automatically re-computed. The slider on the left allows to set the

[3] http://www.wakayama-u.ac.jp/~kawahara/MatlabRealtimeSpeechTools/
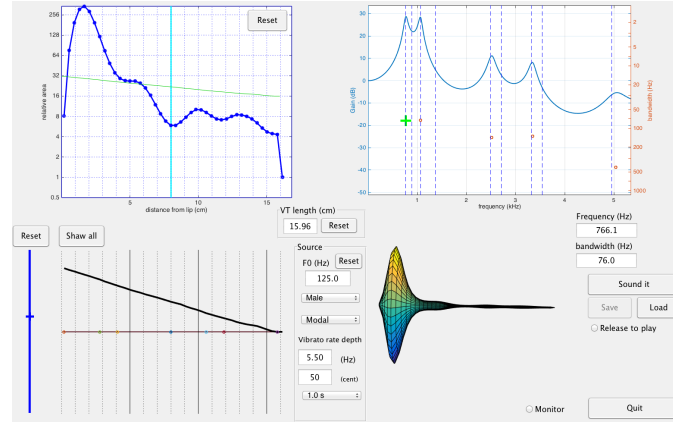
FIGURE 6.2: Interface of the sparkNG software
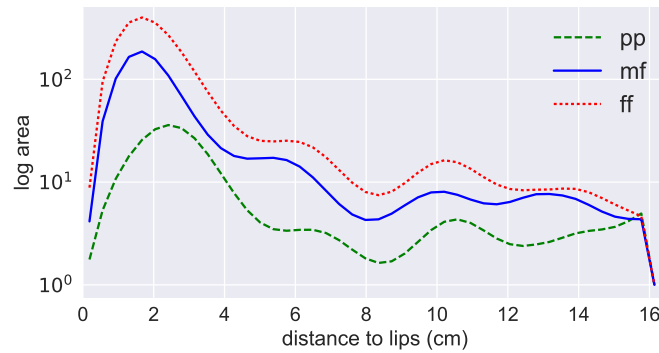
FIGURE 6.3: Estimated VTS for vowel /a/ at intensities *pp*, *mf*, *ff*

degree of this modification. A button also allows to synthesize the corresponding sound (the source being generated based on the LF model), giving an idea of the perceptual effect related to the modification.
Based on the analysis exposed in the previous section, we used the estimated all-pole models to get the corresponding VTS. Figure 6.3 shows as an example the estimated VTS for the vowel /a/ sung at 3 different intensity levels (*pp*, *mf* and *ff*). As could be expected, it clearly exhibits an increasing mouth opening from *pp* to *ff*. Similar results can be obtained for other open vowels, but not for closed vowels like /u/.
For the rest of our experiment, we thus only used the 4 most open French vowels: /a/, /E/, /9/ and /O/. As a first approximation of the VTS change induced by mouth opening, we used a linear slope from the glottis to the lips, as shown in figure 6.2 (bottom left), and applied it to the estimated VTS of the 4 vowels sung at medium intensity level (*mf*). By scaling this shape modification curve using the slider, such that the opening at the lips was multiplied by factor $\gamma \in [0.25, 0.5, 1., 2, 4]$ (on a linear scale), we could then measure the variations of the formants parameters induced according to the degree of mouth opening.

Results:
Figure 6.4 shows the ratios of the estimated formants frequencies (R_{F_i}) after modification of the shape over the original values, for the 4 vowels (with γ displayed on a log scale). Similarly, figure 6.5 shows the ratios for the estimated bandwidths
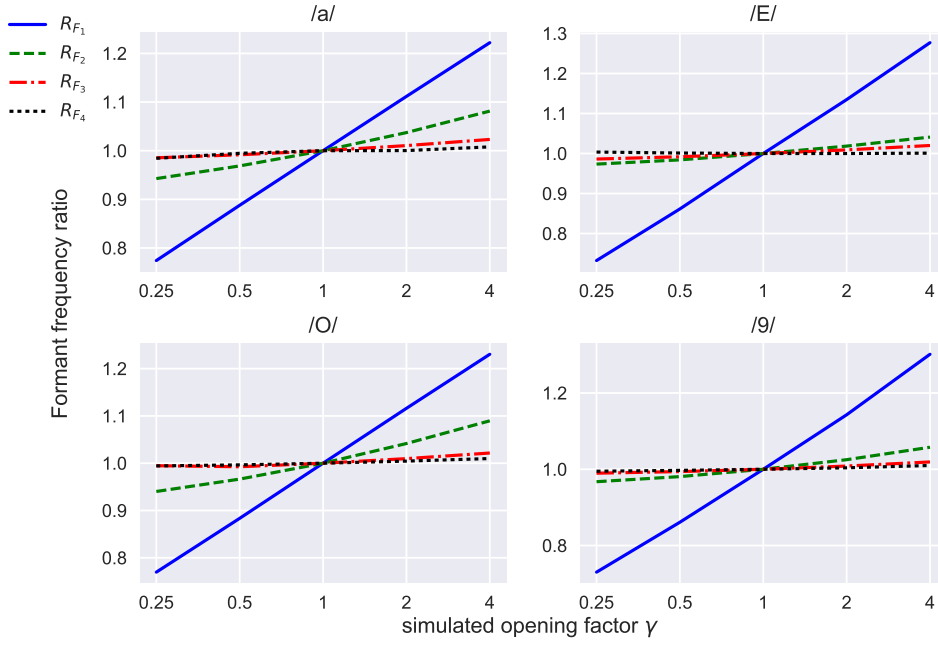
FIGURE 6.4: Ratios of new and original formants frequencies as a function of γ (in log2 scale) for vowels /a/, /E/, /O/ and /9/

(R_{BW_i}). As one can see, the main effect of the simulated mouth opening is a linear increase of F_1 and linear decrease for BW_1 (according to $log2(\gamma)$), similarly for the 4 vowels. Comparatively, the effect on the 3 other formants is negligible. Again, this increase of F_1 is coherent with previous studies and observations, and this correlation between F_1 and jaw opening was also already clearly illustrated in figures 2.2 and 2.3. But the decrease of the 1st formant's bandwidth has rarely been reported in the literature. A decrease of all formants' bandwidths with intensity was however reported in [Mol+14], and [Par02] also evoked an increase of the 1st formant's bandwidth in soft breathy voices, which is thus coherent with the results of the present simulation, as soft voices are assumed to be related to a small mouth opening and thus a larger bandwidth.

6.2.2.3 Transformation procedure

Rule for formant's modification:
From these simulations, a simple rule has been inferred, based on the mean slope for F_1 and BW_1 over the 4 simulated vowels. This rule is given by the following 2 equations:

$$F_{1_{new}} = F_1 \cdot (1 + 0.25\alpha) \tag{6.5}$$

$$BW_{1_{new}} = BW_1 \cdot (1 - 0.4\alpha) \tag{6.6}$$

where $\alpha \in [-1; 1]$ is the opening factor to control the degree of the transformation, and $F_{1_{new}}$ and $BW_{1_{new}}$ are the new values of F_1 and BW_1 to be applied for transforming the original sound. The gain of the formant was not included in this rule, as it is directly linked to its bandwidth and to possible interactions with other close formants.
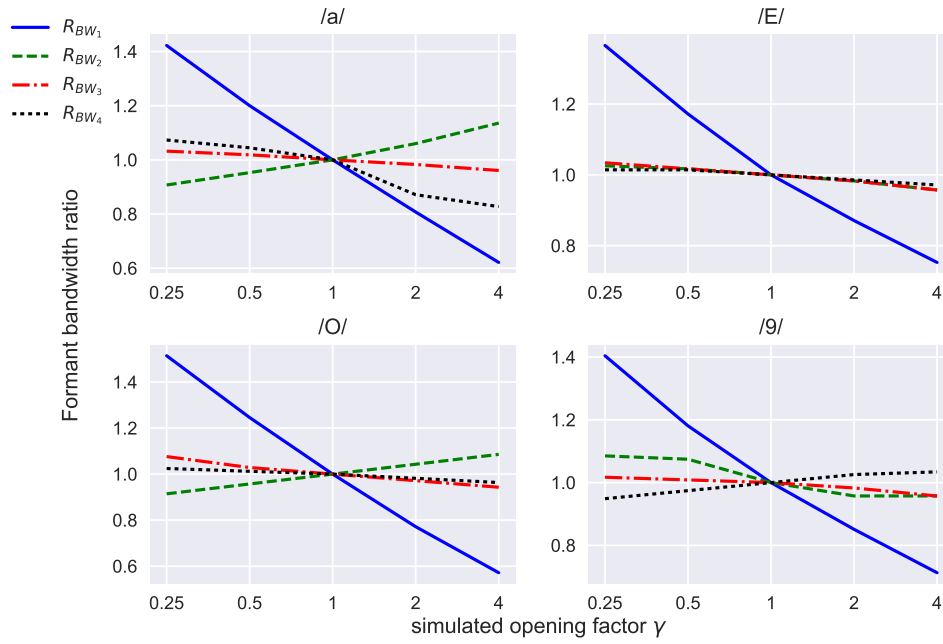
FIGURE 6.5: Ratios of new and original formants bandwidths as a function of γ (in log2 scale) for vowels /a/, /E/, /O/ and /9/

Proposed transformation approach:

In order to apply the defined rule, the spectral envelope has to be modified appropriately to include the change in F_1 and BW_1. This can be done easily in formants synthesizers, but applying this effect on any real voice sample is more challenging.

In [Don+11], frequency warping is used in order to transform the spectral envelope to create natural pitch changes in singing voice. However, it is not easy to precisely change the formants parameters using frequency warping. We first tested this possible approach, but the perceived effects (according to informal listening) remained limited. Especially, the natural increase of the amplitude of a formant when its bandwidth decreases is not reproduced, and formants can't be merged together when getting closer to one another as they should.

In [Mol+14], the authors used a parametric model of spectral envelope based on 4-pole resonators (somewhat similar to Klatt's formant synthesizer [Kla80]) to modify gain, spectral tilt, and formants' frequencies and bandwidths, based on regressions computed from 60 recorded vowels. However, this approach requires to properly extract the parameters of all formants, which is not straightforward without manual correction. Another drawback in [Mol+14] is that the regressions are computed from all vowels, without considering the gender of the singer or the type of vowel. This probably oversmoothes the variations and does not allow to observe the typical move of F_1 related to mouth opening, as mouth opening can't physiologically be as prominent for closed vowels like /i/ or /u/ than for open vowels like /a/.

Another possible approach that we propose to use here is poles modification, based on an all-pole model of spectral envelope. This possibility is mentioned in [San+16], but has been discarded for being too complicated. A similar approach was used in [MAH93] and [HC96] to modify formants, focusing on controlling poles interaction. These works are also mentioned in [Lee05], but the author evokes as a limit that the amplitude and bandwidth of the formants can't be
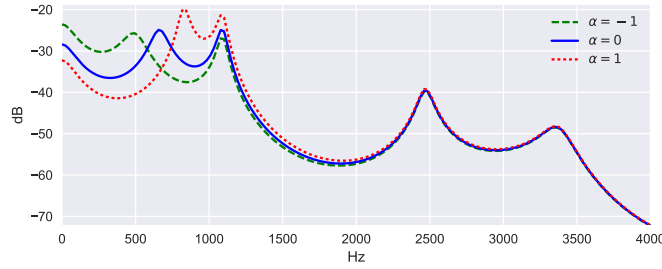
FIGURE 6.6: Transformation of vowel /a/ with $\alpha \in [-1, 0, 1]$

controlled independently. However, we assume that this is not necessary, as amplitude should naturally change when bandwidth is modified, as stated in [Par02]. The proposed method is described below.

Considering an all-pole model of the spectral envelope, assuming that the model is estimated with an appropriate order, the most important formants of the voice should be associated with a pair of conjugate poles.
From the AR coefficients (the a_k in equation 6.2), the roots r_i of the polynomial can be computed by factorization. Then, the frequencies and -3dB bandwidths of the poles, in Hz, are given by the following formulas [Lee05]:

$$F_i = \frac{f_s}{2\pi} \angle r_i \tag{6.7}$$

$$BW_i = -\frac{f_s}{\pi} ln(|r_i|) \tag{6.8}$$

Once the frequency and bandwidth of each pole are known, the values corresponding to the 1st formant can be selected and modified according to equations 6.5 and 6.6. From there, the equations 6.7 and 6.8 can be inverted in order to get the new roots of the model. Finally, these modified roots are converted back to the AR coefficients of the new transfer function, using Leja ordering to limit effects of rounding errors in computation [Ped11]. Figure 6.6 shows an example of using this poles modification approach to apply the given rule on a spectral envelope of the vowel /a/, for $\alpha \in [-1, 0, 1]$.
The transformation is then applied by inverse filtering the original sound by the estimated spectral envelope, and filtering it back with the newly modified envelope.

Note that for the mouth opening effect, the conversion back and forth between the acoustic tube model and the AR coefficients could theoretically allow us to apply the effect by modifying the vocal tract area function and converting it directly to AR coefficients to generate a new envelope. But the proposed procedure has the advantage to:

- allow the direct observation of the changes in formants parameters to deduce a simple rule that can then potentially be used in any type of synthesis system (parametric, concatenative, ...), whereas it relies on an all-pole model or not;

- allow to apply other rules for formants modifications, that are not necessarily related to a specific or well-known change of shape of the vocal tract (e.g. for formant tuning as proposed in [HSW11]).

6.2.2.4 Evaluation

Finally, an evaluation has been conducted to validate the effectiveness of the proposed rule and approach. For this purpose, 2 online listening tests have been run for evaluating both the quality of the transformed sounds in term of naturalness, and the proper perception of the degree of mouth opening produced by the effect. A demo page with the sounds used in the tests can be found at url[4]. All sounds were normalized at the same level, and listeners had to use headphones or earphones to do the tests.

Original sounds:
For the 2 tests, recordings of the vowels /a/, /E/, /9/ and /O/, sung by both a male (RT) and a female (MS) professional singer, at a fixed pitch (135Hz for RT, and 250Hz for MS) and a medium intensity (*mf*), were selected. On each of these sounds, the transformation was applied with $\alpha \in [-1; -0.5; 0; 0.5; 1]$, 0 meaning no transformation. A total of 40 sounds (8 original sounds × 5 α values) were thus used in the tests.

AR model estimation:
For the transformations, an all-pole envelope model was first estimated for each sound, using an implementation of the DAP algorithm [EM91] in SuperVP. An important parameter to be considered for this analysis is the order of the model. From the results presented in [VRR07; RVR07] (about order selection of all-pole models) and informal tests, we used an order of 50 which seemed appropriate in our case.

Estimation of the 1st formant:
Ideally, the pair of conjugate poles with the lowest absolute frequency should correspond to the 1st formant, but depending on the analysis, this may not always be the case. For a more robust estimation of the 1st formant, we thus imposed as a constraint that $F_1 \in [350 - 1000]Hz$. However, this range could potentially be adapted according to the analyzed voice and vowel (which are known in our concatenative synthesis system). Then, the pair of conjugate poles with the lowest frequency following these constraints is selected.

Stable frames selection:
For transforming a sound, the pole modification has to be applied on each frame. Neighbouring frames should have similar formants and therefore pole parameters. However, the analysis may sometimes contain some jumps in the estimated poles' frequencies from one frame to another, which could create artifacts. To avoid this, the median value of the F_1 over the central stable part of the vowel is computed. Then, only the 25% of the frames for which the estimated F_1 are the closest to the median are kept. The other frames are discarded and the remaining ones are interpolated (after modification of F_1 and BW_1), in order to fill the gaps.

Test 1: Quality of the transformation:
In the first test, 15 of the 40 sounds where randomly selected and presented in random order, with no repetition. The sounds then had to be rated on a 1 to 5 scale by listeners following a standard MOS test procedure [ITU16], according to the

[4] http://recherche.ircam.fr/anasyn/ardaillon/mouthOpening2017/demo.php
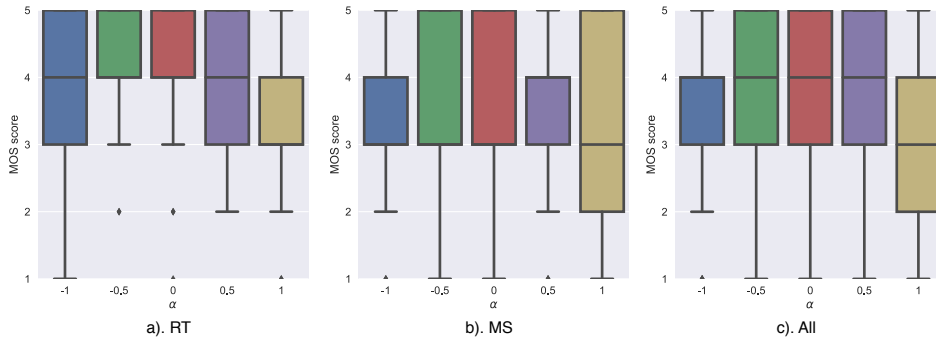
FIGURE 6.7: MOS test evaluating the quality of the transformation according to α, for both voices RT and MS and all sounds confounded

perceived naturalness of the sounds.
48 listeners answered this test. Figure 6.7 shows the results for the different α values for both voices RT and MS and for all sounds. As one can see on figure 6.7 c)., the answers highly overlap and span the whole [1-5] range for all sounds categories, with no difference between α values -0.5, 0, and 0.5, when considering all sounds together. The mean rating is however a bit lower for higher transformation levels -1 and 1. Possible explanations for these slightly lower results may be that the degree of the effect applied (α) starts to be a bit too important to stay fully realistic, or that the changes in F_1 may reinforce too much some particular sinusoid which may then become a bit to prominent and start to whistle. The results were slightly better for RT than for MS, which is understandable as the lower pitch of RT eases the spectral envelope estimation compared to MS. From those results, we can thus assume that the proposed approach can apply the transformation appropriately without important degradation of the naturalness and sound quality.

Test 2: Perception of transformed mouth opening:
The second test presented 15 pairs of sounds to the listener. For each pair, one voice (male or female) and one vowel where first randomly selected, and 2 different sounds were randomly chosen among the 5 possible transformation levels to form the pair. Then, for each pair, a grade from 0 to 3 had to be given according to which sound seemed to be related to a more widely open mouth and how big was the perceived difference between the 2 sounds, following a standard CMOS procedure [Rec03], using a similar web interface to that shown in figure 4.20.
Before starting the test, listeners were given, as a reference, sounds of a vowel /a/ recorded with 5 degrees of mouth opening (from maximally closed to maximally open), in order to give an idea of the expected perceptual effect on a real voice. Also, as the transformation may sometimes change the perception of the vowel identity (an /a/ pronounced with a closed mouth may sound almost like an /o/ or /2/), the transformed vowel was written above each pair of sounds.
36 persons answered this test. As one can see on figure 6.8, those results perfectly reflect the expected effect of the transformation. None of the confidence intervals overlap, which means that the difference between each degree of the applied effect is clearly perceived by listeners, and the results were very similar for both voices. As a result, one can assume that the proposed rule and transformation method are effective for simulating mouth opening.
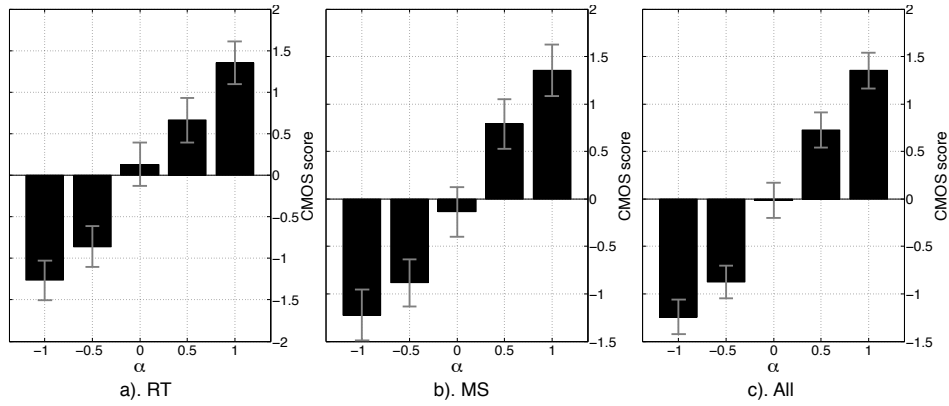
FIGURE 6.8: CMOS evaluation of the perceived degree of mouth opening induced by the transformation, according to α, for both voices RT and MS and all sounds confounded

6.2.3 Loudness correction

The transformations proposed above aim at modifying the voice timbre according to intensity. But the final intensity level obtained after these transformations (and possibly others like transposition) can not be easily predicted. In the previous chapter, we proposed an approach to generate a target intensity level, which should then be matched at the end of the synthesis, once all sound transformations have been applied. For this purpose, a correction gain should thus be computed in order to adjust the final level.
The computation of such a gain requires a certain measure of the intensity, in order to assess the difference between the target intensity level and that of the synthesized voice. A possible simple measure of a signal's intensity is the RMS (Root Mean Square). However, this measure is only related to the amplitude of the temporal waveform of the signal and doesn't reflect well the intensity level that is really perceived by listeners. In order to better assess the perceived intensity of sounds, loudness models have been developed [FPR11], which should thus better be used for the control of the synthesis rather than RMS. According to [FPR11], loudness can be defined as *"the perceptual strength of a sound that ranges from very soft (or quiet) to very loud"*, or *"the subjective intensity of a sound"*, that is closely associated with measures of acoustical level (energy, power, or pressure) but not identical to any of them. This definition suggests that there is unfortunately no objective measure of loudness, which only exists "within" a listener and depends on many factors (e.g. frequency, bandwidth, duration, spectral complexity, presence of other sounds, age of the listener, etc...). There is thus no perfect approach for measuring loudness, which can only really be assessed subjectively for each listener, although *"most people behave in a consistent manner when judging loudness"*. However, several models have been proposed to approximate the loudness level of a given sound from measurements [FPR11]. But the human auditory system is complex and advanced loudness models may be computationally too heavy for our purpose, and often specific to certain types of sounds or listening conditions. We thus propose here to use a simplified loudness model for the control of intensity, based on informations from [FPR11], that we present below.
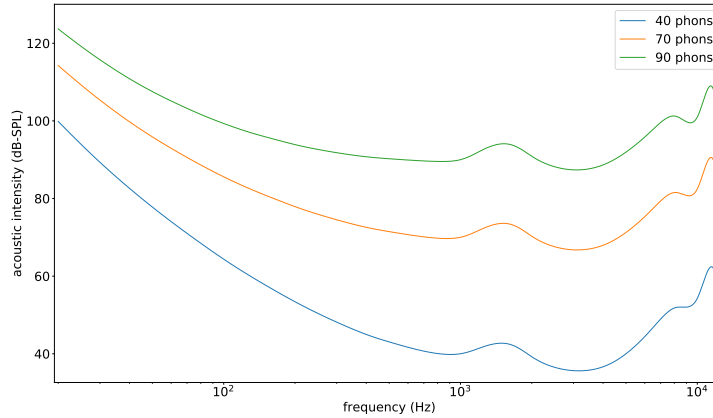
FIGURE 6.9: equal loudness curves for loudness levels of 40, 70, and 90 phons

6.2.3.1 Simple loudness model

Background:
The unit commonly used for measuring loudness is the sone, which is defined in [FPR11] as *"the loudness of a 1-kHz tone at 40dB-SPL heard binaurally in a free field from a source in the listener's frontal plane"*. The sone scale is linear, which means that a sound with a loudness of 2 sones is perceived as twice as loud as a sound with a loudness of 1 sone. The absolute value of loudness in sones, relying on an arbitrary choice, is thus not important (only the relative variations are). Among the many factors that may influence the perception of a sound, the loudness of pure tones mainly varies as a function of intensity (obviously) and frequency.

Regarding the relation between loudness and sound intensity, most studies agree that, for a sound whose intensity level is comprised between 40dB-SPL and 90dB-SPL (which is a reasonable assumption for human voice), the loudness varies as a function of intensity (measured in dB-SPL) following a power law with an exponent around 0.3 (which is commonly known as the Stevens' power law).

Regarding frequency, the equal loudness contours, defined by the ISO226 standard [ISO03], give the intensity values in dB-SPL required to maintain an equal loudness (for a pure tone) as the frequency shifts over the entire audible range. The unit of loudness level used for defining those curves is the phon, where a loudness level of X phons corresponds to the loudness of a 1-kHz pure tone at X dB-SPL. Figure 6.9 shows the equal loudness curves for 3 different phon values. The 3 contours in this figure pass respectively through 40, 70, and 90dB-SPL at 1 kHz and the points on those contours are thus said to have loudness levels of respectively 40, 70, and 90 phons.
The conversion from the loudness level L_N in phons into the loudness N in sones can then be obtained with the following equation 6.9:

$$N = (10^{(\frac{L_N - 40}{10})})^{0.3} \tag{6.9}$$

Until now, we have been considering only pure tones, which is the most simple

case for the computation of loudness. The question then is how the loudness of each individual component contributes to the overall loudness of a complex sound. According to [FPR11], the results of past studies can often be summarized by stating that energy sums for components that are in the same critical band, whereas loudness sums for components that are in different critical bands. Models like that presented in [Zwi60] assume that loudness can be first determined separately in each critical band and then summed across all critical bands (defined by the Bark scale [Zwi61]).

Other factors, like the type of sound, the durations, and the temporal or spectral masking, have implications on loudness measurements. But such factors are not considered here for the sake of simplicity.

Implemented model:
Based on those informations, the simple loudness model we propose to use here for singing voice mainly relies on 4 assumptions:

– The sound is considered to be steady (which is really verified only for sustained vowels without vibrato).

– The sound level is >40dB-SPL (and <90dB-SPL) so that the SPL to sones conversion curve follows a simple power low with exponant 0.3.

– The voice signal is considered to be a simple sum of sinusoids (the impact of the unvoiced part on loudness being thus assumed to be negligible).

– The harmonics are assumed to be sufficiently sparse, lying in different critical bands (which is true at least for the 6 first harmonics, for an f_0 value >90Hz), so that spectral masking effects are limited and the global loudness can be obtained as a simple summation of the specific loudness of each harmonic.

Based on those assumptions, the loudness is computed as follows:

– First, a sinusoidal model is built from the synthesized voice, using the harmonic partials analysis evoked the previous chapter (based on [Bon+11] and [Röb08]).

– Then, the sinusoids amplitudes are converted to phons, based on the equal loudness contours. We assume for this purpose that the average level of voice is around 70dB-SPL and thus use the curve corresponding to 70phons (shown in figure 6.9) for this conversion. This conversion is then done by adjusting the harmonics amplitudes according to the difference between the value of the curve (in dB-SPL) and the reference phon value (70), so that frequency regions that contribute less to loudness are attenuated.

– Then, the specific loudness of each harmonic, in sones, is computed according to equation 6.9.

– Finally, the global loudness, in sones, is obtained by a simple summation of the specific loudness over all harmonics.

This process can be summarized by the following equation 6.10:

$$\begin{aligned} L_{Ni} &= 20\log_{10}(a_i) - (ELC_{70}(\omega_i) - 70) \\ N &= \textstyle\sum_{i=1}^{K}(10^{(\frac{L_{Ni}-40}{10})})^{0.3} \end{aligned} \tag{6.10}$$

where N is the global loudness of the sound in sones, a_i is the amplitude of the i^{th} harmonic, ω_i is the frequency of the i^{th} harmonic, $ELC_{70}(\omega)$ is the equal loudness curve corresponding to a loudness level of 70phons, and K is the total number of harmonics. Note that the values a_i should theoretically correspond to the sound pressure (relative to a reference level of 20μPa corresponding to the threshold of hearing) so that $20\log_{10}(a_i)$ gives a value in dB-SPL. However, the value a_i depends on the distance and sensibility of the microphone and on the gain of the recording, as well as the amplifier and speakers used to listen to the sound, which are unknown. But this is of no importance, because applying a fixed gain to the values a_i would result in a simple fixed scaling of the measured loudness, which does not affect the relative loudness variations in which we are interested.

Although all the assumptions of this model are not completely fulfilled, we assume that this measure should already be closer to the reality of perception than simpler measures like RMS, while still allowing a rather simple and efficient computation. A particular advantage of this simple model is also that it easily allows to deduce a correction gain to be applied to a sound in order to match a given target loudness value, as will be explained below in section 6.2.3.3.
Figure 6.10 compares the loudness estimated with our simple model with that of an implementation of Zwicker's model [FZ90] (from the Ircam Descriptors [Pee04], also available in audiosculpt), and with a simpler short-term RMS measure, for 2 files of the RT database. As can be observed, the loudness variations obtained with our simple model match well the curves obtained with the Zwicker's model, but the RMS measure results in more important variations than the loudness measures. In particular, in the second case, one can observe that the RMS is almost divided by 2 between the 2 vowels, whereas the loudness values are in fact slightly increased, which demonstrates well why the loudness measure should better be used for the control of intensity rather than the RMS.

6.2.3.2 Dependence of loudness on vowels

During the recording of our synthesis databases, singers were asked to sing at a constant intensity level, as previously explained in section 3.2 (although a precise monitoring is very feasible). However, by analysing the loudness values of on our databases with our model, it appears that the measured loudness still varies a lot, although the singer is supposed to keep a stable level. Figures 6.11 and 6.12 show the distribution of the loudness values computed on each vowel of the RT and MS databases, and normalized by the mean loudness value measured on the vowel /a/ (which has the highest values). Similar results are obtained with the Zwicker's loudness model. An interesting observation from those figures is that the most closed vowels have the lowest measured loudness, while the most open ones have the biggest values. Note that the vowels in those figures have been approximately ordered by degree of openness, from the most open to the most closed vowels.

Although the measured loudness are different, the perceived loudness are however relatively similar among the vowels (according to informal listening), such that imposing the same target loudness values to all vowels during the synthesis results in differences of perceived loudness between vowels when listening to the result, which is not what is expected. This is illustrated with sound 6.2 for which all vowels have been synthesized with a similar target loudness level, although the perceived loudness varies between vowels. To circumvent this, it is thus necessary
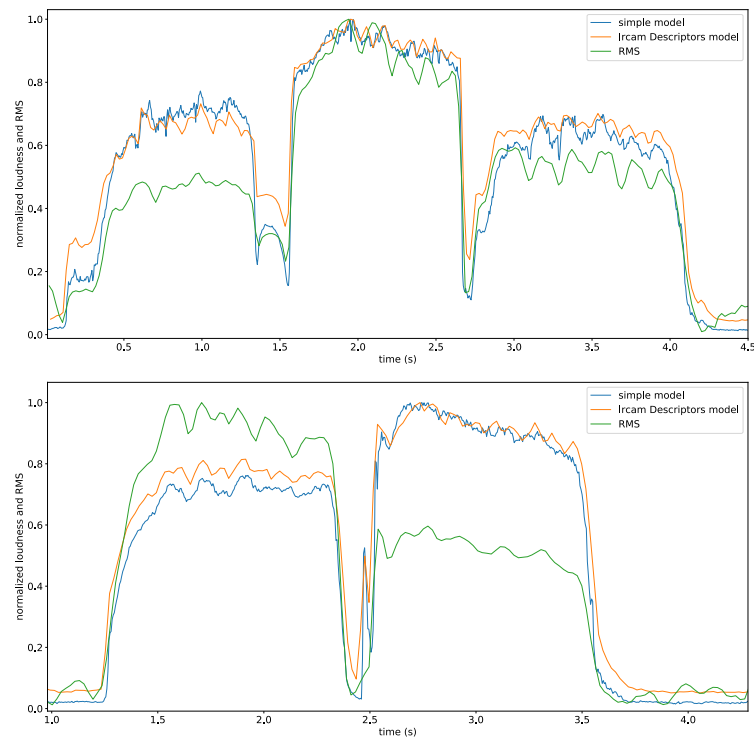
FIGURE 6.10: Comparison of the normalized loudness computed with our loudness model and Zwicker's model, and the normalized short-term RMS, for 2 sounds of the RT database
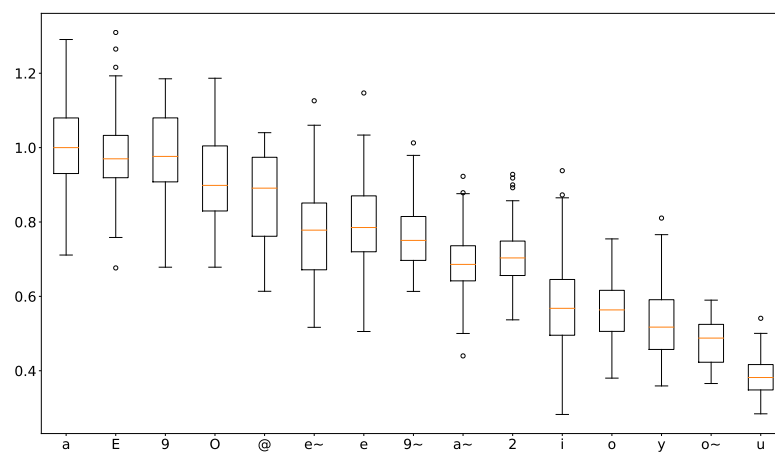
FIGURE 6.11: Distribution of measured loudness of each vowels on RT database (the values have been normalized by the mean value of the vowel /a/)
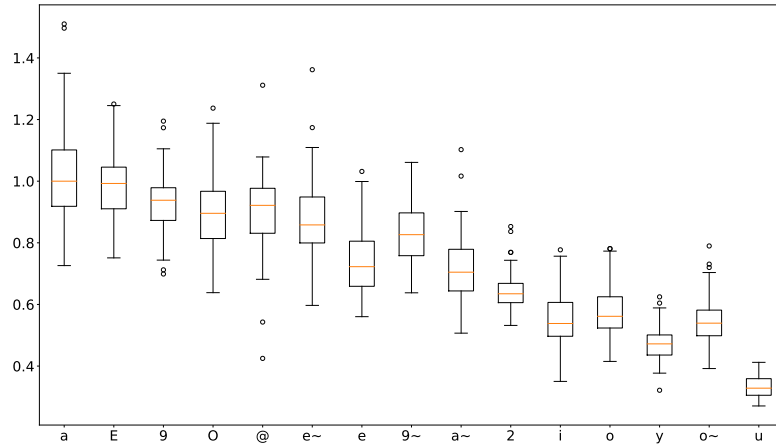
FIGURE 6.12: Distribution of measured loudness of each vowels on MS database (the values have been normalized by the mean value of the vowel /a/)
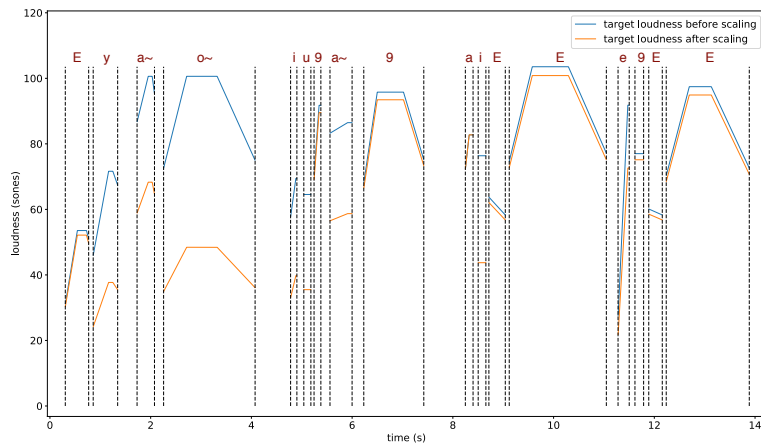
FIGURE 6.13: Example of target loudness curve generated by the control module, before and after re-scaling the curve according to each vowel (the sung vowel is written above each note)

to rescale the target loudness values according to each vowel. This is done by multiplying the loudness profile of each vowel generated by the control module by the mean normalized value of the corresponding vowel measured on the database (as shown in figures 6.11 and 6.12). In Sound 6.3, such vowel-dependent correction has been applied, and the perceived loudness is more homogeneous compared to sound 6.2. Figure 6.13 shows an example of a target loudness curve generated by the control model, before and after this correction.

The differences between the measured loudness and the loudness really perceived may be explained by the fact that there exists some perceptual effects specific to vocal sounds that are not taken into account by the loudness model. An hypothesis is that our perception is more sensible to the vocal effort, and different vowels with similar vocal efforts would thus be perceived with a similar loudness, independently of the vowel, while the proposed loudness measure is more sensible
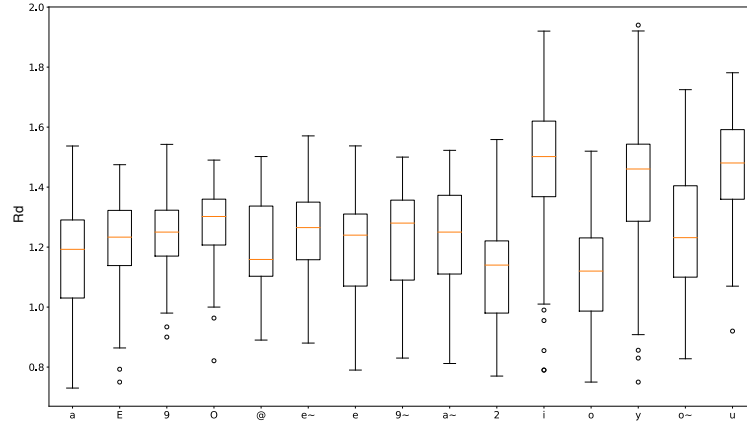
FIGURE 6.14: Distribution of measured R_d values of each vowels on RT database

to the differences in formants between the different vowels. Figure 6.14 shows the distribution of the R_d parameter analyzed on the RT database for each vowel. As can be observed, the R_d value, strongly related to vocal effort, is relatively stable, with very few differences between vowels, which tends to support our hypothesis. However, loudness is easier to measure than vocal effort and thus more convenient to use for controlling intensity.

6.2.3.3 Correction gain

Based on the proposed loudness measure, we need to compute a correction gain α to be applied to the synthesis as a last step, in order to match the target loudness curve generated by the control module (once re-scaled properly according to the sung vowels).

Denoting N_t the target loudness and N_s the loudness of the synthesis before this loudness correction, we have (at a specific instant t):

$$\begin{aligned} N_t = \gamma N_s &= \textstyle\sum_i (10^{\frac{20\log_{10}(\alpha a_i)-(ELC_{70}(\omega_i)-70)-40}{10}})^{0.3} \\ &= (10^{\frac{20\log_{10}(\alpha)}{10}})^{0.3} \textstyle\sum_i (10^{\frac{20\log_{10}(a_i)-(ELC_{70}(\omega_i)-70)-40}{10}})^{0.3} \\ &= (10^{\frac{20\log_{10}(\alpha)}{10}})^{0.3} N_s \end{aligned} \tag{6.11}$$

As N_t and N_s are known (N_s being computed on the synthesized sound just before loudness correction), we have $\gamma = \frac{N_t}{N_s}$, and the correction gain α can thus be computed, from equation 6.11, as:

$$\alpha = 10^{0.5\log_{10}(\gamma^{\frac{1}{0.3}})} \tag{6.12}$$

Using this correction gain, the final loudness should match exactly the target value. Note that such a correction gain could not be obtained so easily using more complex loudness models.

Figure 6.15 shows an example comparing the loudness measured on the original
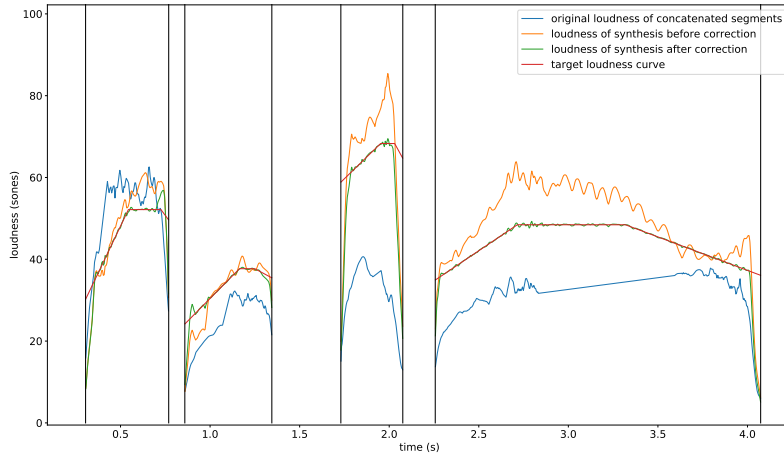
FIGURE 6.15

concatenated segments, the loudness of the synthesized voice before loudness correction (with the rule described in section 6.2.1 for glottal source modification applied), the target loudness curve, and the loudness of the synthesis after correction. Only the vowels segments are shown, delimited by the vertical lines, the gain being linearly interpolated in-between. As can be seen, the corrected loudness matches the target curve, as expected. (Note that a small margin of 0.06s is used at the vowels' boundaries to compute the correction gain, in order to avoid amplifying too much the sound at voice onsets and offsets.)

6.3 Vocal roughness

As evoked previously in sections 2.5.4 and 5.2, certain singing styles, such as pop, blues, rock, etc..., make use of some specific expressive timbre effects related to vocal roughness. We already reviewed in section 2.5.4 the possible physiological causes of vocal roughness, as well as the main signal's characteristics, and the state-of-the-art approaches to model such effects. The term "vocal roughness" is quite general and may encompass different voice qualities that may be identified as rough, but the vocabulary is lacking to precisely name each of those possible voice qualities. In [Nie08], the author proposed a classification of rough voices into 5 different categories, which are: rattle, distortion, growl, grunt, and scream. But the terms used tend to vary in the literature and it is thus easier to describe such voice qualities in terms of their associated spectral and temporal signal's characteristics.

From the signal point of view, we propose here to simply classify rough voices into 2 broad categories:

- One for voices whose main spectral characteristic is the presence of clearly identifiable sub-harmonics that are rather stable in time (with few bifurcations). This type of voices is often referred to as "growl" (or "growl-type")

The work presented in this section was partly supported by the CREAM project: http://cream.ircam.fr/ and conducted in collaboration with Dr. Marco Liuni.

[LB04; Sak+04; Nie08; BB13], but may also encompass the rattle and distortion effects described in [Nie08]. An example of spectrogram for this category is shown in figure 6.16.

- One for other more unstable (or chaotic) regimes, where sub-harmonics may still be visible in the spectrum, but are more unstable with possible bifurcations and with the presence of noise between the harmonics. An example of spectrogram for this category is shown in figure 6.17. This type of voices can also be described as "harsh".

More extreme types of effects may be defined, for very noisy voices which don't even have an f_0 anymore, but we don't aim here at modeling such extreme types of voices, which are too distant from the modal voices used in our databases for singing synthesis.
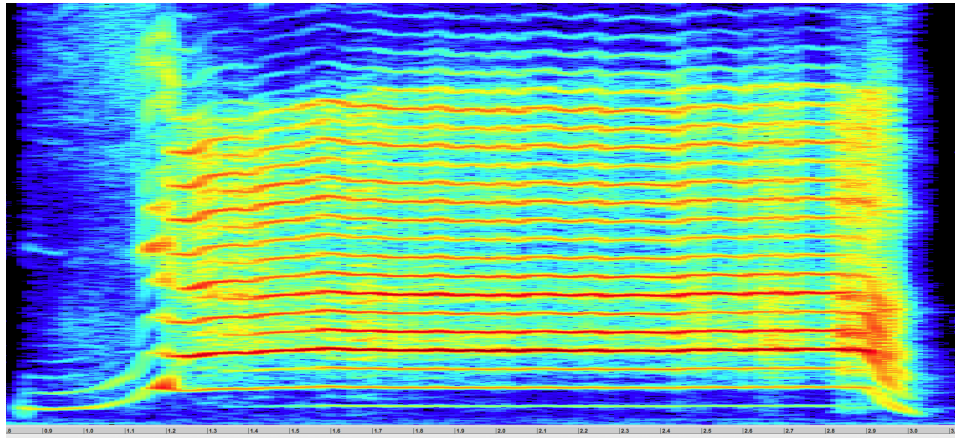
FIGURE 6.16: Spectrogram of a sound from the 1st category (growl effect) with stable sub-harmonics
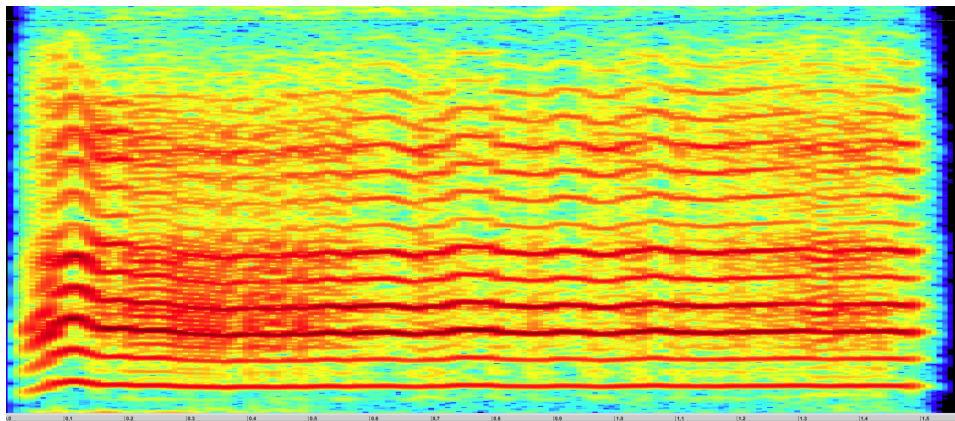
FIGURE 6.17: Spectrogram of a sound from the 2nd category with unstable sub-harmonics and noise

We present in this section two different approaches to apply such voice qualities from those 2 categories on modal (or "clean") voices. As rough voices are usually related to a rather high vocal effort, those effects should however better be applied on voices that are already tense or loud to obtain natural results. For softer voices, the intensity transformation discussed above could be used as a first step

before applying the roughness transformation.
Our first approach is based on a simple amplitude modulation and time-domain filtering to efficiently create sub-harmonics in the original signal, suitable mainly for voices of the 1st category. The second approach is based on the PaN synthesis engine, introduced in section 3.5.2 to generate jitter and shimmer by modifying the glottal pulses' positions and amplitudes, capable of modeling more chaotic behaviours characteristic of the sounds from the 2nd category. (Note that, in this section, jitter denotes very local pulse-to-pulse f_0 variations, whereas the jitter in section 4.4.4 tended to designate longer-term variations.)

6.3.1 1st approach: amplitude modulation

Amplitude modulation simply consists in multiplying in the time-domain a carrier signal with another one (the modulating signal) with a lower frequency and an amplitude comprised in the range [0-1], centered around 1.
Let $x_c(t) = A_c \cos(\omega_c t)$ be the carrier signal, with an angular frequency ω_c and an amplitude A_c, and $x_m(t) = 1 + h\cos(\omega_m t)$ be the modulating signal with an angular frequency ω_m and a modulation depth $h \in [0-1]$ (also called the modulation index). We then have, as a result of the modulation:

$$\begin{aligned} y(t) &= x_m(t)x_c(t) \\ &= (1 + h\cos(\omega_m t))A_c\cos(\omega_c t) \\ &= A_c\cos(\omega_c t) + A_c h\cos(\omega_m t)\cos(\omega_c t) \\ &= A_c\cos(\omega_c t) + \frac{A_c h}{2}\cos((\omega_c+\omega_m)t) + \frac{A_c h}{2}\cos((\omega_c-\omega_m)t) \\ &= x_c(t) + y_+(t) + y_-(t) \end{aligned} \tag{6.13}$$

The resulting signal thus contains the original sinusoidal carrier signal $x_c(t)$, and 2 new sinusoids $y_+(t) = \frac{A_c h}{2}\cos((\omega_c+\omega_m)t)$ and $y_-(t) = \frac{A_c h}{2}\cos((\omega_c-\omega_m)t)$ with amplitudes $\frac{A_c h}{2}$ at frequencies $\omega_c + \omega_m$ and $\omega_c - \omega_m$.

Now, let's consider $x_c(t)$ being a voice signal, approximated by a simple sum of N harmonic sinusoids: $x_c(t) = \sum_{i=1}^{N} A_i \cos(i\omega_0 t)$ (where $\omega_0 = 2\pi f_0$). The result of the modulation of this signal by $x_m(t)$ would simply be the sum of each harmonic modulated individually:

$$\begin{aligned} y(t) &= x_m(t)x_c(t) \\ &= x_m(t)\sum_{i=1}^{N} A_i\cos(i\omega_0 t) \\ &= \sum_{i=1}^{N} x_m(t)A_i\cos(i\omega_0 t) \\ &= x_c(t) + \sum_{i=1}^{N}(y_{+_i}(t) + y_{-_i}(t)) \end{aligned} \tag{6.14}$$

with $y_{+_i}(t) = \frac{A_i h}{2}\cos((i\omega_0+\omega_m)t)$ and $y_{-_i}(t) = \frac{A_i h}{2}\cos((i\omega_0-\omega_m)t)$.
By choosing an appropriate value for ω_m, it is thus possible to generate sub-harmonics between each harmonics at frequencies $i\omega_0 \pm \omega_m$, the distance of each sub-harmonic to its related harmonic i being thus equal to ω_m. Thus, setting $\omega_m = \frac{\omega_0}{k}$, this would generate a pair of sub-harmonics around each harmonic at $i\omega_0 \pm \frac{\omega_0}{k}$.
A particular case is $\omega_m = \frac{\omega_0}{2}$ where the upper sub-harmonic generated by the ith harmonic and the lower sub-harmonic generated by the (i+1)th harmonic have the same frequency ($i\omega_0+\omega_m = (i+1)\omega_0-\omega_m$). This results in a single sub-harmonic being generated between each pair of harmonics (as can be often observed on real signals like the one from figure 6.16).

It is also possible to use a sum of sinusoids for the modulating signal, in order to generate more sub-harmonics:

$$x_m(\omega_0, t) = 1 + \sum_{k=1}^{K} h_k \cos\left(\frac{\omega_0}{k} t\right) \tag{6.15}$$

For instance, in order to generate 3 equally-spaced sub-harmonics, one may use the sum of 2 sinusoids at $\omega_0/2$ and $\omega_0/4$.

Note that in terms of signal's characteristics, temporal amplitude modulation can be related to some kind of shimmer (in this case with a regular periodic pattern and not random variations, as the modulation frequency is directly related to the f_0). Sub-harmonics may also be obtained using frequency modulation (which would then be rather related to jitter). In [Gio+99], the author states that such non-linear combination of 2 signals with amplitude and phase modulations produce lateral waves and relates this phenomena as an evidence of coupling between the 2 vocal folds. However, frequency modulation generates an infinite series of sub-harmonics (lateral waves) with more complex amplitude relations, which are thus more complex to control for our purpose.

Figure 6.19 b) shows the result of such an amplitude modulation on an original "clean" voice signal, whose spectrogram is shown on figure 6.19 a), for the simple case where $\omega_m = \frac{\omega_0}{2}$. However, using only this modulation doesn't result in a natural-sounding voice signal. The reason for this is that the amplitudes of the lowest sub-harmonics (and especially that of the first one, below the fundamental) are too high. Observing real signals, such as the one shown in figure 6.16, we can see that the amplitudes are much lower for the lowest sub-harmonics.
In order to obtain something similar, it is thus necessary to high-pass filter the sub-harmonics. As the original signal $x_c(t)$ is fully preserved in the modulated signal, the generated sub-harmonics can easily be isolated by simply subtracting this original signal from the modulated one: $y_{sub}(t) = y(t) - x_c(t)$. Once the sub-harmonics have been isolated, they can be high-pass filtered before being added back to the original signal by a simple summation. We use for this purpose a butterworth digital IIR high-pass filter. We thus obtain the final rough voice signal as:

$$y_{rough}(t) = x_c(t) + \alpha y_{sub}^{HP}(t) \tag{6.16}$$

where $y_{sub}^{HP}(t)$ denotes the high-pass filtered sub-harmonics, and $\alpha > 0$ is a mixing factor. The whole algorithm is summarized in figure 6.18, and an example with the signals obtained at each step of the algorithm is shown in figure 6.19.

Figure 6.19 e) shows the final result of the effect, after the filtering step. In this example, we tried to obtain a result similar to the real voice shown in figure 6.16. For this purpose, we used for the modulation a single sinusoid at $\omega_0/2$, with a modulation depth $h = 0.75$, a 3$^{\text{rd}}$ order filter with a cut-off frequency at 1000Hz, and a mixing factor $\alpha = 1$.
However, it seems, from observations, that those parameters, and especially the filter's cut-off frequency, tend to vary from one voice to another, giving different qualities of roughness. From the physiological point of view, the amplitude modulation may possibly be related to the interaction between the vocal folds and other vibrating supra-glottal structures such as the ventricular folds. For instance in [Bai09], the author observed vibrations of the ventricular folds at frequencies
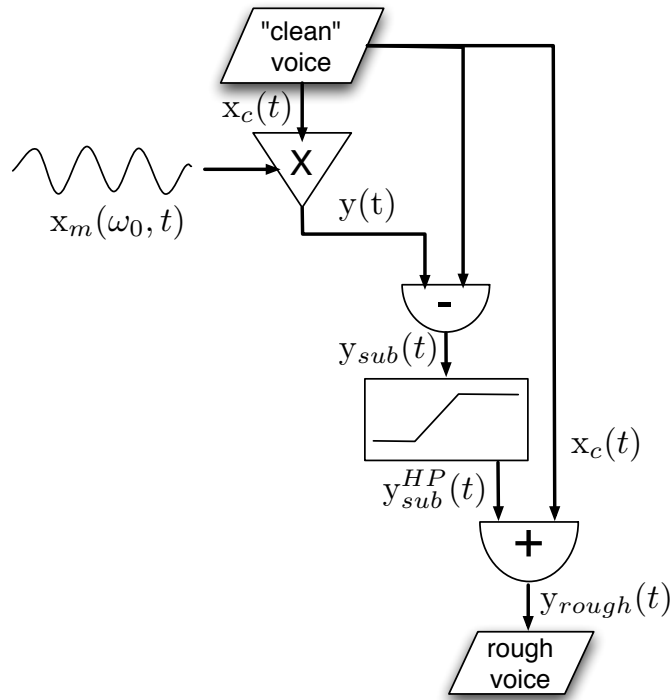
FIGURE 6.18: Schematic of the amplitude-modulation-based roughness algorithm

$f_0/2$ or $f_0/3$, which may then modulate the original sound wave generated by the vocal folds. But it remains unclear how the high-pass filtering of the sub-harmonics relates to the voice physiology, and if the cut-off frequency should be related to the f_0 or rather be fixed according to other physiological factors (e.g. vocal tract configuration). The modulation index could also be changed to obtain a more or less intense effect. Another simple way of varying the intensity of the effect is to vary the mixing factor α.
Using appropriate settings, this simple approach has proved to give very natural results on several examples. But due to the lack of time, no proper evaluation has been done yet, and a subjective listening test should be conducted in order to evaluate the naturalness of the sounds produced using a more extensive set of voice samples and define several presets for generating different rough qualities.
According to our observations on various recordings, it appears that real voices may contain from 1 to 5 sub-harmonics. The more sub-harmonics there are, the more rough the voice sounds. Figure 6.20 shows another example of the effect applied on the same original voice, but using as modulating signal a sum of 3 sinusoids at $f_0/2$, $f_0/3$ and $f_0/6$, with a modulation depth of 0.5, $\alpha = 1$, and the same filter as above. Some sounds for the 2 examples illustrated in figures 6.19 and 6.20 and others can be found on the web page at url[5] (sounds 6.4 to 6.15).

Although other approaches have been proposed to generate sub-harmonics to create roughness in voice (e.g. [Nie08; LB04; BB13]), the main advantage of this approach, beyond its simplicity and the naturalness of the results obtained, is its efficiency. The only operations required to apply this effect are 1 multiplication for the amplitude modulation, a subtraction to isolate sub-harmonics, a few

[5] http://recherche.ircam.fr/anasyn/ardaillon/these/these.php
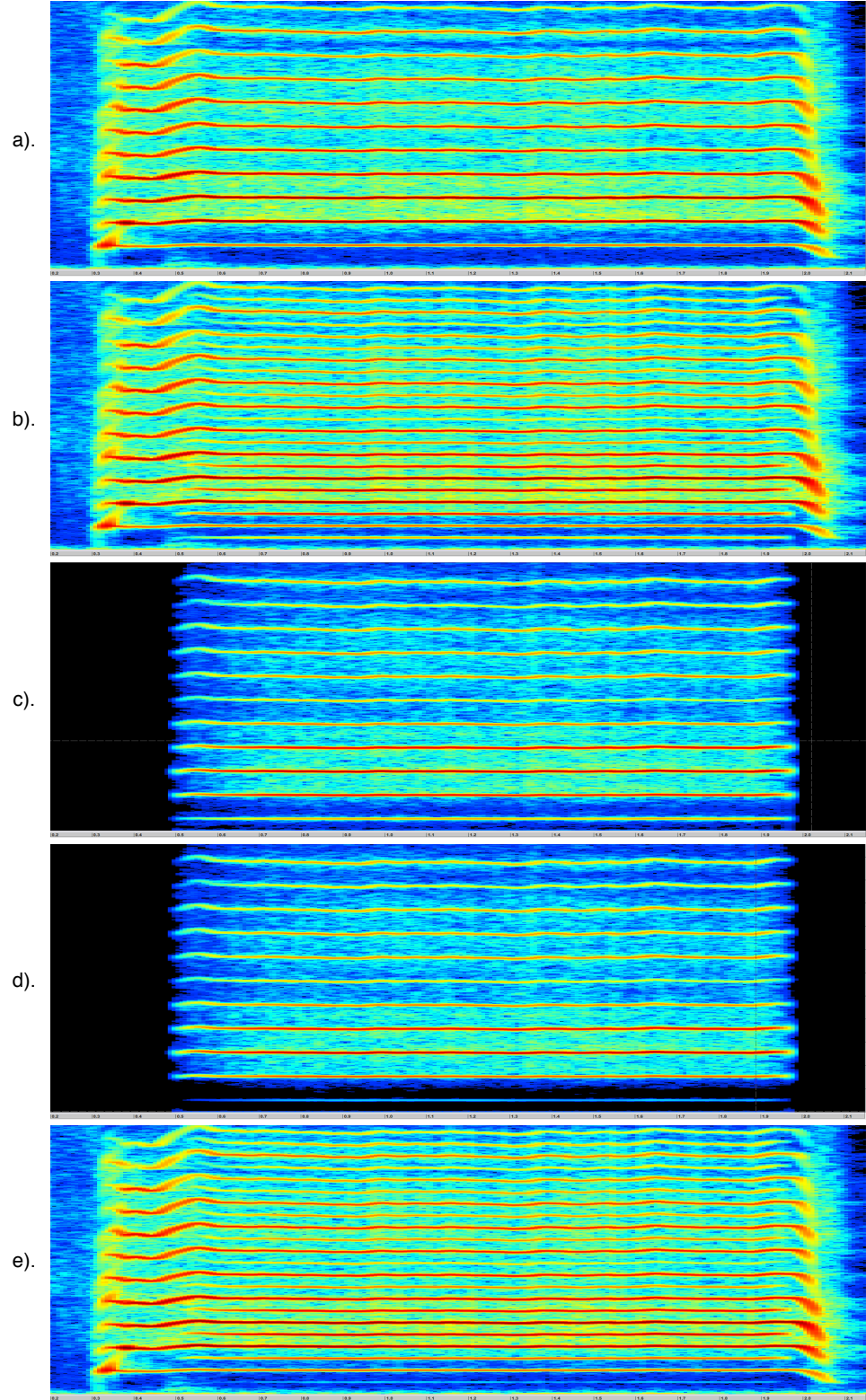
FIGURE 6.19: Example of the amplitude-modulation roughness effect, showing the spectrum of signals at each step of the algorithm. a). original "clean" voice signal $x_c(t)$; b). amplitude-modulated signal $y(t)$; c). isolated sub-harmonics $y_{sub}(t)$; d). high-pass filtered sub-harmonics $y_{sub}^{HP}(t)$; e). final rough voice signal $y_{rough}(t)$
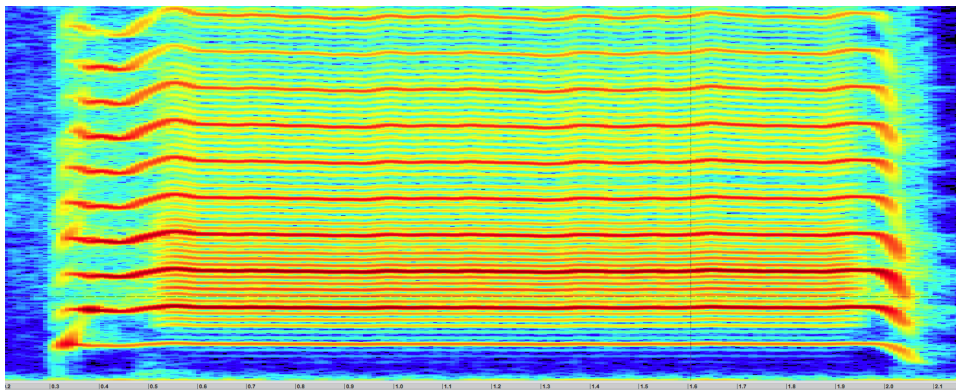
FIGURE 6.20: Example of roughness effect with 5 sub-harmonics, using a sum of 3 sinusoids for the modulating signal

multiplication and additions for the filtering (depending on the order of the filter), and an addition and a multiplication for the final mixing step. This thus makes this approach especially suitable for real-time. Although real-time is not necessary for its integration into our singing synthesizer ISiS, this nevertheless allows other interesting applications of this effect, which may be very easily integrated into real-time singing synthesizers like [Feu+17], or used as an audio effect on a real voice, for instance for a singer who is not able to produce this effect himself. The effect has first been developed in python, from which a max/MSP implementation has been derived for real-time, thanks to the work of Dr. Marco Liuni. The most computationally-heavy step for applying this effect in real-time is however the f_0 estimation, that is necessary for setting appropriate frequencies for the modulating signal. We used for this a real-time implementation of the yin algorithm [CK02], available at url[6], which allowed us to implement the effect with no audible latency (note that for real-time synthesis systems like [Feu+17], this f_0 estimation would not even be necessary, as the f_0 value is directly provided by the user).

The presented approach is especially suitable for generating sounds with stable sub-harmonics, from the first category defined above. However, for more unstable types of rough voices, the number of sub-harmonics and the modulation parameters could be changed along time to create bifurcations between different regimes. Another possibility would be to use not only sinusoids for the modulating signal, but also more chaotic signals, e.g. using band-pass filtered noise. However, more research would be necessary to properly investigate those possibilities.
Finally, another remaining open question about this approach is the influence of the phase of the modulating signal. From empirical testing, it appeared that this phase value may influence the quality of the result (at least for the single sub-harmonic case). Some more tests would thus be necessary to investigate this question and determine if some refinements may be necessary to properly control the phase of the modulating sinusoids (e.g. to better align them according to the pulse positions).

6.3.2 2nd approach: jitter and shimmer modeling with PaN

In the previous section, we focused on the spectral characteristics of rough voices from the 1st category, proposing a simple approach to generate stable sub-harmonics using amplitude modulation. But for voices of the 2nd defined

[6] http://forumnet.ircam.fr/product/max-sound-box-en/
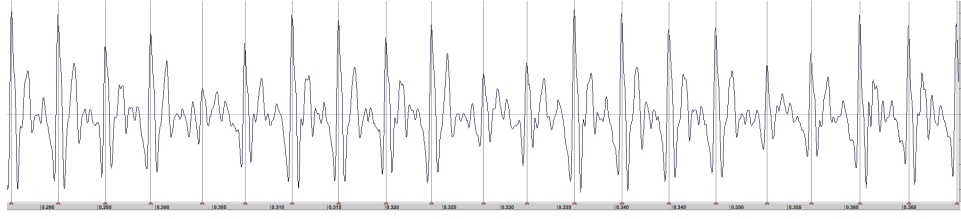
FIGURE 6.21: Waveform extract of a rough voice (from the same recording as figure 6.17) with annotated periods
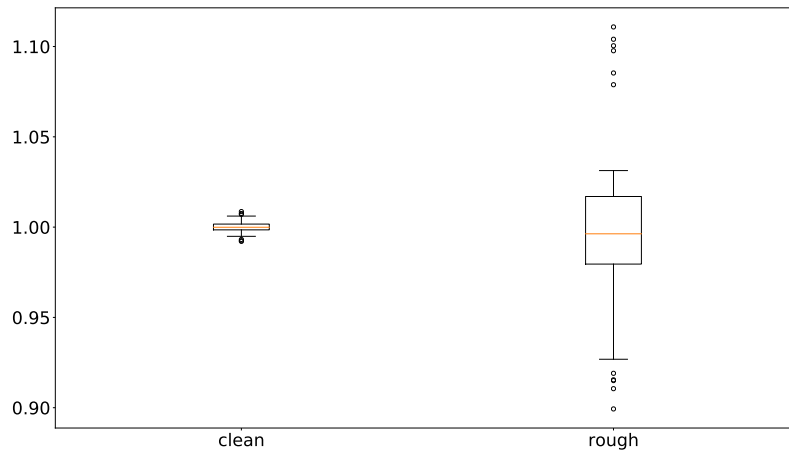
FIGURE 6.22: Distribution of ratios of individual periods over the periods obtained form the low-passed f_0 of 2 analyzed segment from both a "clean" and a rough voice extracts from the same singer, showing the presence of jitter in the rough voice

category, sub-harmonics are not always present, or are less stable, and signals are mainly characterized, in the time domain, by the presence of jitter and shimmer, that can be described respectively as (pseudo-)random pule-to-pulse frequency and amplitude variations, as has been attested in previous studies [BB13; VK05; Jon+01].
Figure 6.21 shows an extract of the waveform corresponding to the spectrogram of a rough voice from figure 6.17, along with markers at the peaks of each glottal cycles. As one can see, the peak amplitude varies from one glottal pulse to another (while the VTF is assumed to be relatively stable in this example). It is not easy to visualize the jitter directly on the waveform, but figure 6.22 shows the distributions of the ratio of the local period of each glottal cycle, obtained from the markers shown in figure 6.21, over the period obtained from the low-passed f_0 analysis at the positions between each markers, both for this rough voice sound and another "clean" voice sample from the same singer. Figure 6.23 shows the superposition of the annotated glottal cycles on the same segments, normalized in duration (using resampling). As one can see, the waveform is much more constant for the clean voice than for the rough voice.

Using the PaN parametric synthesis engine presented in section 3.5.2, we generate each glottal pulse individually and thus have the possibility to precisely control their positions and amplitudes. We thus propose here to model rough
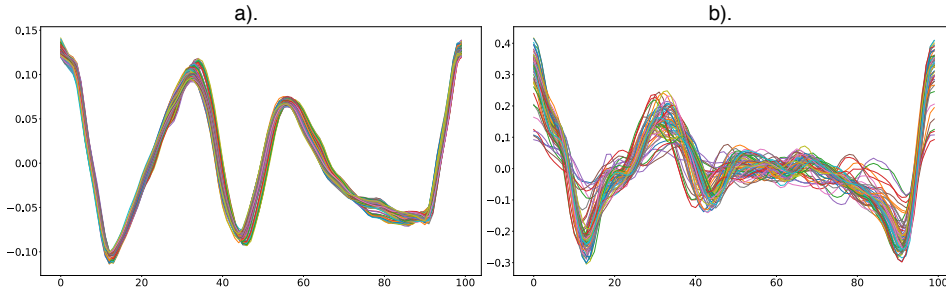
FIGURE 6.23: Superposed normalized periods for a "clean" (a) and a rough (b) voice
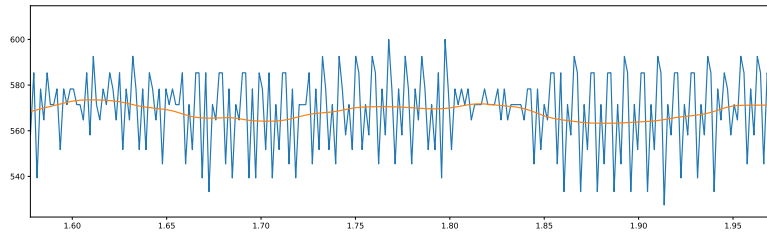
FIGURE 6.24: Jitter extract showing the local f_0 values for individual glottal cycles and the low-passed f_0

voices by introducing jitter and shimmer by controlling individual pulses (as was already suggested as a perspective in [Bon04]), using the PaN engine, based on the analysis of real rough voices.

Currently available f_0 estimation methods (such as the one used in our synthesis system) may be disturbed by the presence of sub-harmonics and are not accurate enough to precisely extract the local frequency at each glottal pulse to estimate jitter. It is thus necessary to use a more accurate approach for extracting the pulse-to-pulse frequency variations in rough voice samples. The R_d estimation algorithm presented in section 2.2.4 can give an estimate of the glottal closure instant of each glottal pulse, along with the estimated R_d value. However, this algorithm first requires an estimate of the f_0. In order to precisely analyse jitter and shimmer from rough voices recordings, we thus first run an approximate estimation of the f_0 and manually correct it when necessary. Then, the R_d estimation algorithm is used to get an estimate of each glottal pulse position. But this algorithm assumes that the voice source fits a usual shape that approximately matches the LF model, which may not be the case for rough voices and can lead to estimations errors. The positions estimated by this algorithm are thus moved to the biggest peak of each glottal cycle around the initially estimated position. Finally, those positions are verified and manually corrected when necessary. These peaks positions then allow us to precisely extract the frequency and amplitude of each individual glottal cycle. An example of such annotation was already shown in figure 6.21. Then, from this annotation, the jitter and shimmer can be estimated by computing the ratio of local f_0 and amplitude values estimated for each glottal cycle over a low-pass filtered version, as shown on figures 6.24 and 6.25 for jitter, and stored as templates.

Then these templates, stored as simple factors centered around 1, can be used during synthesis to rescale the original periods and amplitudes of each glottal
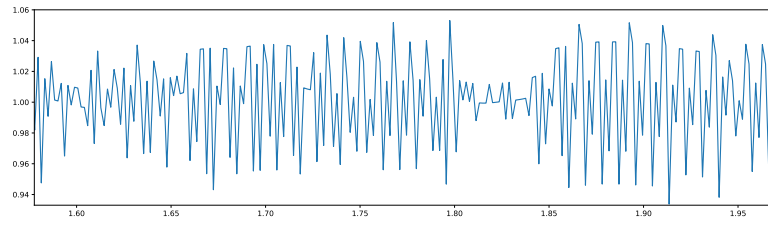
FIGURE 6.25: Jitter template as ratio of local frequency over low-passed version
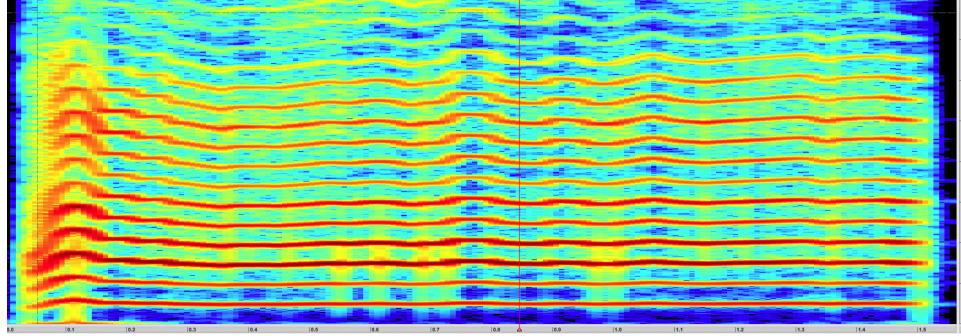
FIGURE 6.26: Resynthesis of sound from figure 6.17 with the PaN engine after suppression of jitter and shimmers

cycles of a pre-analyzed voice before resynthesizing it using the PaN engine, as previously described in section 3.5.2. Using this approach, the degree of jitter and shimmer analyzed on a real rough voice can thus be varied to modify the rough quality of the voice, or transposed on another voice to introduce roughness, possibly using looping to extend the extract, if necessary, as is done for morphing-based approaches [BB13].

Figure 6.26 shows the resynthesis of the sound from figure 6.17 where the jitter and shimmer have been removed by low-pass filtering the f_0 and spectral envelope to produce a clean version. In Figure 6.27, the original shimmer and jitter patterns extracted from the original rough voice (such as the template shown in figure 6.25) have been reintroduced in the PaN synthesis by scaling the glottal pulse periods and amplitudes according to these patterns. As can be seen on this figure, the spectrum is very similar to the original one, contai ning noise and sub-harmonics, and the result thus also sounds rather close to the original rough voice.

An advantage of this approach over other approaches like spectral morphing [BB13] is that besides using real recordings, it is also possible to build artificial jitter or shimmer patterns from the ground up without necessarily requiring to analyze real recordings. It also gives more control over the behaviour of the jitter and shimmer to vary the degree and type of roughness. Note that more regular (or periodic) patterns can also be used to generate stable sub-harmonics. For instance, repeating similar jitter factors every 2 cycles generates 1 sub-harmonic between each harmonic (spaced by $f_0/2$), and using a repetition rate of 3 glottal cycles generates 2 sub-harmonics between each harmonic (spaced by $f_0/3$), as illustrated in figure 6.28. Such local regularity can also be observed on real sounds as for instance in the patterns from figure 6.25. This approach could thus also be used to generate rough sounds with stable sub-harmonics from the 1^{st} category.
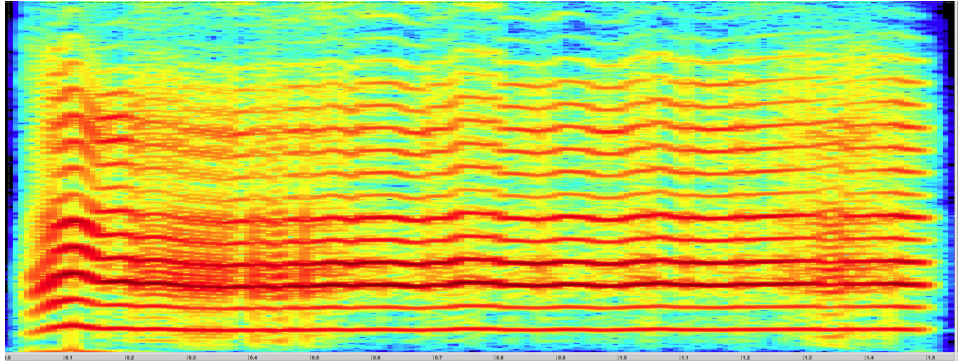
FIGURE 6.27: Resynthesis of sound from figure 6.17 with the PaN engine, applying the jitter and shimmer patterns extracted from the original sound
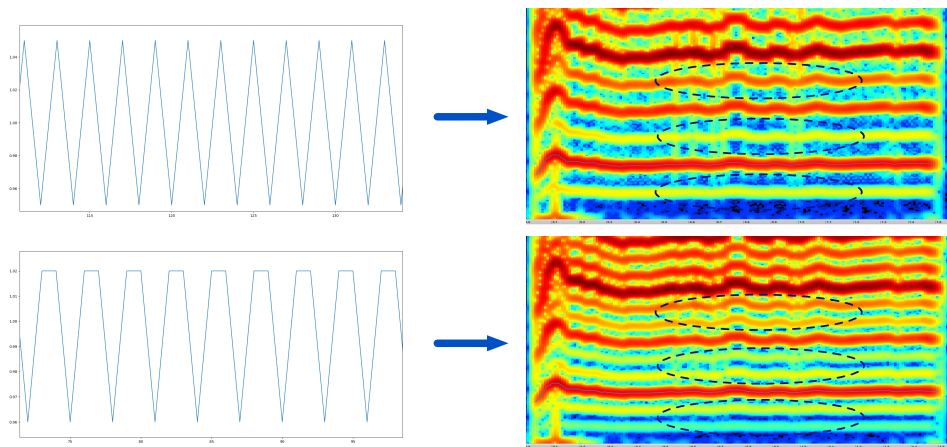
FIGURE 6.28: Example of using regularly alternating jitter factors. Sub-harmonics are generated in the spectrum depending on the alternance rate.

Some examples of results obtained using this 2nd approach are given in sounds 6.16 to 6.24.

Using this 2nd approach, we managed to generate some artificial rough voices that sound similar to original recordings. This can thus be accounted as a proof of concept for applying roughness transformations by introducing jitter and shimmer with the PaN engine. However, we haven't managed yet with this approach to obtain results that sound as natural as the original rough sounds we analyzed. An hypothesis to explain this is that besides the frequency and amplitudes of the glottal cycles, their shape should also change from one period to another. Small variations of the waveform can indeed be observed for instance on figure 6.21. A possibility would be to apply a filter that varies from one glottal pulse to another, as has been suggested in [Bon04] and [Nie08], but further research would be necessary to investigate this possibility. Another limitation is that, contrary to our 1st approach based on amplitude modulation, it is not possible here to filter the sub-harmonics which may be too important in the low frequency range. But an hypothesis is that what we observe as jitter and shimmer should rather be assumed to be consequences of a more global change of the shape of the glottal pulses due to the interaction between the glottal source and other structures, and a possible idea to

simulate this, while limiting the amplitudes of the lower sub-harmonics, would be to introduce some kind of frequency-dependant jitter using all-pass filters to apply different phase shifts depending on the frequency (and possibly use an additional equalization to modify the amplitudes).

6.4 Summary and perspectives

In this chapter, we presented some research aiming at improving the naturalness of the synthesized voices and extending the expressive possibilities required to synthesize various singing styles, by means of timbre transformations. In particular, we first focused on the problem of intensity transformation by modifying both the glottal source and vocal tract filter. Then we presented 2 new approaches to introduce roughness in voice.

For modifying the glottal source, we proposed to use the PaN synthesis engine to modify the R_d parameter along with the energy of the glottal pulses, based on a simple rule. However, preliminary perceptual tests based on this transformation did not allow yet to fully confirm an improvement of naturalness for intensity transformations using this rule alone. Several hypothesis have been formulated to explain this limited result, one of which is that the vocal tract filter should also be coherently modified along with the source component for the effect to be well perceived.
We assume that the most perceptually important modification of the vocal tract for intensity transformations is the mouth (or jaw) opening. We thus proposed to simulate the effect of mouth opening on the voice spectrum. For this purpose we proposed, based on signals analysis and simulations, a simple rule to modify the frequency and bandwidth of the 1^{st} formant. Then, a new approach to apply this rule on real voices has been proposed, based on poles modification from an estimated all-pole model of the spectral envelope. 2 listening tests were conducted in order to validate the effectiveness of the transformation in terms of naturalness of the transformed sounds and of perception of the degree of mouth opening induced. The results of these tests confirmed that the proposed rule and method were very effective for simulating mouth opening on real voice sounds. However, it should be noted that the effect should be adapted to the nature of the transformed vowel, as the degree of mouth opening is physiologically limited for some closed vowels like /i/ or /u/.
As intensity transformation techniques mainly rely on a proper estimation of the spectral envelope, an interesting perspective to further improve the accuracy and robustness of our approach (or for morphing-based approaches) may be to first use the MFA envelope estimation methods mentioned in section 6.2 to derive an all-pole model using the DAP analysis, similarly to the TE-LPC approach proposed in [VRR06] that combines together the True-envelope and the LPC for a more accurate all-pole spectral envelope estimation.
Finally, a means to correct the final loudness to match the target curve from the control module is proposed, by computing a time-varying correction gain, based on a simplified loudness measure that we presented.
This work represents a first step towards a complete parametric vocal intensity transformation. Due to the lack of time, the 2 proposed modifications of the source and vocal tract components have not been combined together yet. Future works should thus investigate the proper tuning of those 2 rules according to vowel,

gender, or singing style, in order to produce the most natural and expressive effect, and a new subjective evaluation should then be conducted. It would also be interesting to compare the results obtained with other approaches such as spectral morphing. The combined effect should then be properly integrated into our ISiS synthesizer. However, other aspects are still to be considered to cover all effects related to vocal effort and intensity. Examples are the noise level, or the singer's formant that can be observed in certain voices depending on gender and singing style. Informal observation on a few recordings also suggested that the zeros of the spectrum in nasal vowels tend to be less prominent with increasing intensity, but this aspect cannot be modeled using an all-pole envelope only and should be further investigated.

While we first focused in this thesis on intensity transformations, the approach proposed for mouth opening transformations could be also similarly applied to pitch transformations. In [Sun90] (p.125), Sundberg illustrated the relation between jaw opening and f_0 for different vowels sung by a professional female opera soprano singer, showing that the jaw opening increases with pitch for all vowels, except for the /a/ (which is already one of the most open vowels). Following this indication, the rule we proposed for simulating mouth opening could thus be used directly for pitch modifications. But other rules could also be applied, using the same approach, for instance related to formant tuning where the frequency of formants are tuned to the fundamental or certain harmonics of the voice, as has been proposed in [HSW11]. Authors in [San+16] also proposed to modify the frequencies of the first 3 formants F_1, F_2 and F_3 for pitch transformations.Our approach could also possibly be used to shift F_3, F_4, and F_5 closer to one another, which is assumed to be the cause of the singer's formant [Bjo08; SLG13].
For pitch modifications, the glottal source component may also need to be modified, as suggested in [DRR11b] which proposed a rule to modify the R_d parameter according to the transposition's factor for speech. This rule has also already been implemented into our synthesizer, but its effectiveness in the case of singing could be further investigated.

Besides our work on intensity, we also proposed 2 approaches for modeling vocal roughness. The 1^{st} approach is based on amplitude modulation and filtering for generating rather stable sub-harmonics characteristic of certain types of rough voices (e.g. growl effect). The advantages of this approach are its high simplicity and efficiency, which make it suitable for real-time transformations. Based on informal listening, we state that this approach is capable of generating roughness on modal voices with a high naturalness. However, this method has not been properly evaluated yet, and subjective listening tests should be conducted to confirm this statement. A remaining open question to be further investigated for this approach is the influence of the phase of the modulating signals on the naturalness of the results.
The second proposed approach to roughness transformation is based on jitter and shimmer modeling using the PaN synthesis engine. While the sounds generated with this technique can reproduce a rough quality similar to that of real sounds that have been analyzed, we did not managed yet to have it sound as natural as the original rough sounds, and the quality of the result tend to vary from one sound to another. An hypothesis to explain this is that the use of the LF-source model in the PaN engine results in a too important regularity of the waveform, despite the applied shimmer and jitter, while each pulse should also probably be

independently modulated or filtered. However, the work presented on this 2nd approach constitutes only a preliminary step mainly aiming at demonstrating the potential of the PaN synthesis engine for such transformations, and further works on this approach will be necessary to improve those results.
Besides singing voice synthesis, these 2 approaches may also be of interest in the framework of perceptual experiments which require to precisely control the characteristics of the stimuli used. In collaboration with the perception team at IRCAM, it is planned to use the presented approaches in the context of experimental studies investigating the perception of emotions in voice and music.

In future work, the modeling of other voice qualities could also be investigated to be used as expressive effects in the context of singing voice synthesis. Examples are breathy or creaky voices. Vocal fry, or more generally the various laryngeal mechanisms, could also be modeled. The PaN synthesis engine offers interesting perspectives for this purpose, as such effects and mechanisms are mainly related to the source component which can be precisely manipulated with PaN. From informal tests, it appears that a simple downward transposition to a very low pitch using the PaN engine already results in a voice quality rather close to vocal fry. In [Bai09], the author states that the closing time of the glottal source (t_a) should be longer for fry (mechanism M0) than for modal voice (mechanism M1), which could be investigated.

Finally, for a proper integration into our singing synthesis system, and for the purpose of singing style modeling, all those timbral effects should ideally be automatically controlled according to the user's inputs, the musical contexts, the chosen voice database, and the target singing style.

Chapter 7

Conclusion

7.1 Summary of personal contributions

The end objective of this thesis was to conduct researches towards the development a high-quality singing voice synthesis system that can, from a given score and lyrics, automatically generate a singing voice that sounds as natural and expressive as possible. This task however implies many aspects, from signal modeling and expressive voice transformations to the control of the synthesis and singing styles modeling, and many problems have to be addressed, that could not be all covered. We summarize in this section the main contributions of this thesis work to the field of singing voice synthesis and transformations.

A thorough state of the art review:
In chapter 2, we reviewed the various aspects related to singing voice synthesis. This review covered the voice production mechanism, the existing synthesis techniques, the approaches to voice modeling and basic transformations (pitch and duration), the possible more advanced voice quality and expressive timbre transformations, the control of the synthesis, and the modeling of singing styles. For each of those aspects, we discussed the main existing techniques, with their advantages and limitations. Although this review is probably not fully exhaustive for all aspects covered, we assume that it gives a good overview of the research field, attesting of the current state of the researches, and pointing out the current limitations.

Development of a state-of-the-art concatenative singing synthesizer:
In chapter 3, we presented our work on the development of a fully-functional state-of-the-art concatenative singing voice synthesizer. We first described the specifications and the process related to the constitution of our synthesis databases. Then, we presented our strategy for the segmentation of those databases. Finally, the implementation of our singing synthesizer *"ISiS"* has been described, along with the signal models and transformations used, and specific processing related to the concatenation to avoid discontinuities at junction between segments. In particular, a first engine based on an advanced phase vocoder implementation (using superVP) has been developed (the "SVP" engine). Then, thanks to the modular architecture of the system, a second parametric synthesis engine (the *PaN* engine, developed by Dr. Axel Roebel) has also been integrated.
Although this work doesn't constitute a major personal contribution in terms of research results, being partly based on previous works and relatively similar to other state-of-the-art concatenative systems, the development of such a system was a first necessary step to identify the limits of the techniques employed, before working on further improvements, and to provide an essential tool allowing to work on other

more specific aspects like expression control and singing style modeling.

A new parametric f_0 model with intuitive controls:
In order to synthesize natural and expressive singing voices from a simple score, providing only notes' pitches and durations, an appropriate control of the synthesis is also necessary. In chapter 4, we addressed the problem of generating the synthesis parameters, encompassing the phonemes durations, the f_0 and the intensity variations.
In particular, we proposed a new multi-layer f_0 model with intuitive controls to generate natural and expressive f_0 variations from the score. This model is mainly composed of a melodico-expressive layer generated using B-splines, with a restricted set of intuitive parameters that can be used to generate attacks, transitions, sustains and releases with expressive fluctuations like preparations, overshoots, and vibrato. To this first layer are also added a jitter and a micro-prosodic component to improve the naturalness of the generated curve.
This model has been evaluated by means of a listening test and has proved to be suitable to generate artificial curves that sound similar to real f_0 curves extracted from recordings of professional singers in different singing styles.

A new approach to automatic tuning of expressive parameters and singing styles modeling:
Although the proposed control model provides intuitive parameters to the user, the manual tuning of the synthesis remains a fastidious task, that should thus better be automatized, which cannot be done without considering the target singing style. In chapter 5, we thus addressed the question of how to automatically generate appropriate parameters to render expressive performances and model various singing styles in the synthesis.
First, we discussed the definition of a singing style, which had not been well established until now in the framework of singing synthesis and tends to vary a lot in the literature, and we reviewed the various aspects implied in its perception.
Based on the suggested definition, we then constituted a stylistic corpus of consistent commercial recordings from 4 famous French singers representative of different singing styles.
Among the various features implied, we proposed in this work to model singing styles based on the 3 basic prosodic parameters that are phonemes durations (not much considered in previous works), the f_0, and the intensity. For this purpose, we developed a new approach to learn the specific variations inherent to each singing style by choosing appropriate parameters estimated on the corpus, according to the target contexts of the score, using decision trees.
Although many aspects related to singing style have not been considered, some encouraging results were obtained in an evaluation of this approach. Besides modeling singing styles, this approach also alleviates the need of manual tuning by automatically generating expressive performances with context-dependant variations, while still providing useful controls to the user to modify the result.

Investigations on parametric intensity transformations, including a new rule and approach to produce a realistic mouth opening effect:
In order to further improve the quality of concatenative synthesizers, one of the main current challenges is to provide appropriate transformations that are coherent with the variations imposed by the control parameters, and extend the timbre space covered by the synthesizer, based on the limited set of recordings contained in the

database.
In chapter 6, we proposed to use a parametric approach for intensity transformations, by decomposing the global effect into several contributions related to different physiological factors. First, we proposed to modify the source spectrum to vary the perceived tenseness of the voice according to the target intensity, by applying a simple modification rule on the R_d and E_e parameters of the LF glottal source model in the PaN synthesis engine.
Then, we proposed a simple rule to modify the frequency and bandwidth of the first formant according to the degree of mouth opening, which is assumed to be related to variations of vocal intensity. Then, in order to apply this effect on real recordings, a new approach has been proposed, based on poles modification of an all-pole model of the spectral envelope. A subjective evaluation confirmed that the proposed rule and approach were very effective to provide a realistic mouth opening effect on sustained vowels recordings. Note that these proposed rule and approach may also be applicable for pitch transformations.
Finally, in order to provide an appropriate control of the sound level according to the target intensity, we proposed to use a simple loudness model, from which a correction gain can be computed to correct the final level of the synthesis.
Although other aspects like the level of aspiration noise, or the singer's formant, could be also considered, we assume that the combination of those 3 components should result in a satisfying intensity transformation, although they have not been properly integrated together in our synthesizer yet.

New approaches to roughness transformations:
In order to further extend the possibilities of the synthesizer and model various singing styles, some specific vocal qualities should also be reproduced. A particular timbral effect used in certain singing styles is vocal roughness. In chapter 6, 2 new approaches to roughness transformations have been investigated.
The first approach we proposed is based on amplitude modulation and time-domain filtering to generate sub-harmonics with appropriate amplitudes in a modal voice recording.
The second approach aims at introducing jitter and shimmer patterns, extracted from real rough samples, in synthesized voices. This is done by controlling the position and amplitude of each glottal pulse using the PaN synthesis engine to reproduce variations similar to that of real rough voices.

7.2 Artistic collaborations

Singing voice synthesis being meant to be used in musical productions, IRCAM offers an ideal environment, by bringing together researchers and composers, to explore the possibilities (and the limits) offered by the results of our research in artistic contexts.
During this thesis, the system developed has been used to produce synthetic voices for the virtual clone of a singer for the opera *"I.D."*[1], composed by Arnaud Petit and written by Alain Fleischer, to be created in october 2017. For this purpose, the EL synthesis database has been used, which has actually been recorded by the soprano singer Eléonore Lemaire, who will interpret the real character of this opera. An extract of the synthesis generated for this piece is given in sound 7.1 (using a temporary midi musical accompaniement).

[1] http://www.lefresnoy.net/panorama18/artwork/710/id/arnaud-petit

Besides this first artistic collaboration, the potential of our approaches to roughness transformations for artistic applications is also being currently explored by the composer Marta Gentilucci in the framework of her composer's residency at IRCAM during the year 2017.
Hopefully, future works will further improve the quality and flexibility of our synthesis system, as well as its accessibility (the current user interface remaining limited), and will spark interest from other composers and artists to explore the possibilities offered by such systems.

7.3 Current limitations and perspectives

In the introduction of this thesis, section 1.5, we listed 3 main challenges to be faced to improve the quality of state-of-the-art synthesizers. In the previous chapters covering the different aspects of singing voice synthesis, some issues related to those challenges have been addressed, and some ideas for future improvements and perspectives have been given. We aim to summarize in this section the main limitations of state-of-the-art systems, including ours, and possible research perspectives to further improve the quality and extend the possibilities of singing synthesis.

Signal modeling and transformations:
Although HMM-based synthesis systems may still be improved, e.g. using better vocoders, their quality currently remains limited due to oversmoothing problems related to the statistical modeling. Neural-network-based synthesizers have recently emerged [BB17] and seem to have good potential to improve the results obtained with HMM-based systems. However, we assume that current concatenative synthesizers still provide the best synthesis examples, regarding sound quality, and can already produce intelligible and rather natural-sounding voices, based on signal modeling techniques like the phase vocoder, or parametric vocoders like PaN. However, such quality can currently be obtained only for a limited range of the control parameters, mainly regarding pitch. The more distant the target parameters are from the original ones, the more artifacts arise. One of the main current challenges is thus related to voice transformations, in order to extend the quality of the synthesis to a larger range of parameters, by producing an homogeneous, natural, and coherent timbre over an important range of pitch or intensity values.

Regarding the pitch transformations, a first cause of artifacts with the phase vocoder, especially for downward transpositions, is related to the amplification of noise in formants after transposition with spectral envelope preservation, which augments the hoarseness of the voice, along with the lack of higher harmonics. In order to limit this effect, the noise could be treated separately from the harmonic part. The use of a parametric glottal source model in the PaN engine, along with a separate processing of the unvoiced part, already tends to improve the quality for downward transpositions over the phase vocoder approach.
But another important limitation for transposition is related to the spectral envelope estimation, especially for high-pitched female voices, due to its sparse sampling by the harmonics. A possible direction to try to improve such estimation, that has been evoked in sections 2.2.3.3 and 6.2, is the use of multi-frame analysis techniques, which has already given encouraging results for transposition, as has been detailed

in [DAR16a], and could thus be further explored. For the PaN approach, the estimation of the R_d parameter also remains difficult for high-pitched voices and should be improved.
Finally, it is also well-known that the spectral envelope is dependant on the pitch, and should thus be modified accordingly. For this purpose, the mouth-opening effect developed in chapter 6 for intensity transformations could also be used in the case of pitch modification (as jaw opening seems to be correlated to pitch, as illustrated in [Sun90], p.125), or adapted to implement other rules related to formants tuning, as suggested in [HSW11]. Besides the VTF, the source component may also need to be adapted according to the pitch, as has been already suggested in [DRR11b], and to model the various laryngeal mechanisms. This possibility should thus also be considered for a fully-realistic transformation.

Regarding intensity transformations, the transformations of the glottal source and spectral envelope proposed in chapter 6, along with loudness correction, should first be properly integrated to create a complete intensity transformation effect, and evaluated. Then, other elements like the level of aspiration noise and the singer's formant should also be investigated and could be integrated in this parametric effect for further improvements. We assume that such a parametric approach could circumvent the limitations of morphing-based approaches, which relies on specific recordings and can't properly model the timbre variations on coarticulation parts, at vowels boundaries. But it would be interesting to compare both approaches in a real use-case with our concatenative synthesizer, once properly integrated, to verify this assumption.

Expression control and singing styles modeling:
Regarding the generation of control parameters, a particular limit of state-of-the-art approaches that we aimed at overcoming in this thesis, while combining the advantages of each technique, is the lack of intuitive means for a user to control and modify the result of the synthesis, especially regarding the f_0. However, the approach we proposed also has its own limitations, the main one being that the possible expressions that can be produced are limited by the restricted set of control parameters provided for tuning the f_0 and intensity curves. All possible contours thus can't be accurately reproduced with our approach, although the proposed models appear to be rather well suited in most cases.

In [Umb15], the author defended the idea of using hybrid approaches to combine the advantages of HMM and units selection-based approaches. In order to further improve our results, while maintaining our idea of providing some higher degree of controllability compared to other approaches, we suggest, as a new perspective, to develop an hybrid approach combining our parametric approach to parameters generation with a units selection-based approach similar to that proposed in [Umb15]. With such idea, real contours could be used to accurately reproduce the expressions of real singers, while still being parametrized and quantified. This way, it would thus still be possible to use decision trees, based on a rich contexts description, for selecting the target units, with the additional possibility to modify the expressive parameters to drive the selection process and further transform the selected units to change the expression.

The modeling of singing styles could then be further improved, by considering more contextual factors in the selection process, like the rhythmical position of

notes in bars and syntactic labels to identify important words to be emphasized.
Regarding the units/templates selection process, it would also be possible to mix the decision tree-based clustering with a cost function similar to that used in units selection, in order to choose the most appropriate units on the leaves of decision trees.
Note that besides the implicit knowledge induced by machine learning, a certain amount of explicit knowledge is still necessary, in order to determine the important contextual factors to be considered (e.g. for building the decision trees, or in the cost function), and to maintain the coherence between the various parameters, using specific rules and constraints (e.g. regarding the alignment between f_0 variations and phonemes, as suggested in section 4.4.2.6).
The coherence between the f_0 and intensity could also be improved, for instance by using the intensity parameters as contextual factors for predicting the f_0, or by using multi-dimensional units, selecting simultaneously the f_0 and intensity units from the same notes in the stylistic corpus.
Finally, other aspects related to singing style should also be modeled, encompassing symbolic features like rhythmical deviations and ornamental notes, and timbral features like voice quality.

Voice quality and expressive timbral effects:
A last challenge to be faced to extend the expressive potential of synthesized voices concerns the modeling of specific voice qualities and expressive timbral effects, necessary to model more varied singing styles. In this thesis, we proposed 2 approaches to produce vocal roughness transformations (such as the growl effect), based on amplitude modulation and jitter and shimmer modeling.
Regarding the 1st approach, the implication of the phase of the modulating signal on the produced sound should however be further investigated.
Although the 2nd approach proved (based on informal listening) to be able to produce voice qualities close to original rough voices in synthesis, further work is still necessary to improve the quality that remains limited in some cases. In particular, besides jitter and shimmer, modulations of the pulse's shape can be observed that are not currently modeled, and the current approach doesn't provide any means to limit the amplitude of the generated sub-harmonics and noise in the low frequencies, which seems important to consider to obtain a natural-sounding result. In future works, we might investigate the possibilities of using all-pass filters and equalization to model the variations on the pulse's shape and limit the amplitudes of the lowest harmonics to improve the quality.
Besides vocal roughness, other timbral features like tenseness or breathiness could also be modeled in the synthesis, for which the PaN synthesis engine has a great potential.
Finally, the automatic control of such effects has not been given much attention until now, and should thus be considered in future works to appropriately generate those vocal effects according to the target singing style and musical contexts.

7.4 About the evaluation of singing voice synthesis

An important, but difficult task for singing synthesis is the evaluation of the results, in order to assess the obtained quality compared to other state-of-the-art systems and to real voices. As there is not much objective measure to assess the quality (considering both naturalness and expressivity) of the synthesized voice,

evaluations mainly rely on subjective listening tests.

As many different aspects are involved in singing synthesis, each one should ideally first be evaluated independently, besides the overall quality of the system, to assess the relevance of the proposed methods. In this thesis, such tests have been conducted for instance for evaluating the f_0 model, the singing styles modeling, and the mouth opening effect, while trying each time to minimize the impact of other aspects.
Depending on the feature to be evaluated, many factors should be carefully considered when conducting a listening test. The type of test to be conducted (e.g: MOS, CMOS, AB, ABX, MUSHRA, ...) is a first element to consider, as different tests may be more or less suited to evaluate certain specific features. In some tests, the sounds are assessed separately, whereas in other tests, they are compared by pairs or groups with other methods and with target or reference sounds. In this thesis, we used for instance the MOS, CMOS and ABX testing procedures to evaluate different aspects, as has been detailed in the previous chapters. The duration of the test and number of sounds to be assessed is also an important aspect to be considered, in order to avoid fatigue and encourage more people to do the test. Some procedures like CMOS or paired comparisons require more judgements from the listeners for a similar number of sounds than other procedures like absolute category ratings (MOS tests). Then, the duration of each sound should also be sufficient to clearly hear the differences in the evaluated features, but not too long to allow a good memorization before rating the sound. Finally, the clarity of the instructions and the precision of the questions asked are of major importance, as depending on how listeners understand it, they might judge different aspects that may not correspond to what was intended.

But such tests may be time-consuming to conduct and require the participation of many listeners, and it is thus not easy to systematically and thoroughly test each aspect of the synthesis.
Furthermore, due to some interdependences, it is sometimes difficult to evaluate certain aspects separately from other ones. It is for instance difficult to evaluate the quality of the control of intensity if the system doesn't already integrate a realistic intensity transformation, because although the control may be appropriate, it might not be judged as such if the transformation doesn't already sound natural. Also, even with well-designed tests and oriented questions, it is sometimes hard for listeners to understand what they are expected to listen, and their attention may be disturbed by artifacts or other aspects that should not be considered.
Besides those aspects, the fact that a listener knows that he/she is listening to an artificial voice tends to change his/her perception, as certain sound attributes that would not be questioned when coming from a real voice may sometimes be considered by listeners as unnatural although they are sometimes similarly present in real voices (e.g. some degree of buzzyness or nasality, the high or very short duration of certain phonemes, small pitch or timbral irregularities, or conversely a very regular sustained vibrato, ...), although this might also be sometimes related to a lack of coherence with other features. For these reasons, the result of a listening test is sometimes difficult to interpret in case it does not correspond to what was expected (as has been evoked for instance in section 6.2.1 regarding our experiment on glottal source transformation with intensity).

In this thesis, we have mainly evaluated the quality of the control of the synthesis, as well as some specific timbre transformations. In order to evaluate the quality of the synthesis (concatenation and signal modeling and transformations) without the influence of its control, a possibility is to use copy synthesis, were all the control parameters like f_0, phonemes durations, and intensity are analyzed on real recordings and given as input to the synthesis. Then, the result can be compared to the original voice (ideally using the same singer in the synthesis database and the target recordings) to identify the remaining artifacts and limitations to be tackled in order to obtain a quality similar to the real voice. It would thus be interesting in the future to do such test with our synthesis system.
It would also be interesting, if possible, to use other singing synthesizers like Vocaloïd[2], Cevio[3], or Utau[4] as different baselines for comparison, as has been done for instance in [Umb15] with Vocaloid for evaluating expression control.

However, the overall evaluation of a synthesis system and its comparison with other ones is a particularly difficult task, due to the disparity and lack of consistency and compatibility between the existing systems, that rely on different synthesis techniques, are targeting different languages, use different databases, provide different interfaces, operate under different constraints (e.g. real-time or offline synthesis), model different features, etc... A list of the past research projects on singing synthesis, and some of the differences between existing systems are given in [Umb+15], in tables 1, 4 and 5. An overview of some subjective and objective evaluations conducted using those different systems is also given in table 10 of this paper. So, unless the different systems have been developed within the same research team or project (as it is the case for instance in [BB17] or [Feu+16]), such comparative evaluation can hardly be conducted.
Some attempts in that direction have however been conducted, at the occasion of 2 singing synthesis challenges at the Interspeech conference, in 2007[5] and 2016[6], the later having been initiated by the collaborators of the ChaNTeR project. In those 2 sessions, a common score was given to be synthesized by the participants with their own systems. In the 2016 session, we participated in the challenge, submitting a song synthesized with the PaN engine and the RT database, using the "Le Roux" style model to generate the expression parameters (cf sound 7.2 and sound 7.3), and our submission was rated 2nd among the 7 submitted synthesis from other participants, based on an online evaluation campaign. However, although this result may give a rough idea on the quality of each system compared to others, it doesn't constitute a rigorous evaluation, for the reasons evoked above, as the database, languages and modeled features differed between the evaluated systems. It is thus hard to know what has really been evaluated for each systems, some being possibly more limited by the signal models used, while others may be more focused on producing a very natural timbre but are more limited regarding the control part.

For a more rigorous comparison between systems, it would be necessary to impose similar conditions for all systems, synthesizing the same score and lyrics

[2] https://www.vocaloid.com/en
[3] http://cevio.jp/
[4] http://utau-synth.com/
[5] https://www.interspeech2007.org/Technical/synthesis_of_singing_challenge.php
[6] https://chanter.limsi.fr/doku.php?id=sidebar

in the same language, using the same database, ideally with similar annotations (as the quality of the synthesis may also be related to the quality of the database annotations). In [Umb+15] and [Umb15], the authors also evoked the necessity of such a common evaluation framework to easily evaluate and compare singing synthesis systems under similar conditions, with a shared evaluation criterion, so that the comparison could focus on the technological differences, independently of the material used. Regarding the testing conditions, several aspects could be shared, like the target songs to be synthesized and the stylistic corpus and databases used for generating the sound and modeling the expressive features, as well as an evaluation framework with clear instructions on what should be rated and how. However, due to the differences between systems, the task of building a publicly available dataset for this purpose may not be so easy, and would need some common agreements among the developers of the different systems on what should be provided.

As a first step in that direction, an evaluation has been conducted, in the framework of ChaNTeR project, to compare the results obtained with our system, with both the PaN and SVP engines, with that of another system (a singing instrument called "Calliphony", controlled in real-time) developed at the Limsi laboratory (collaborator of the ChaNTeR project), and with real singing in several conditions, as has been detailed in [Feu+16]. In this study, 2 conditions were used for generating the synthesis, in order to evaluate the impact of the concatenation process on the final result. For the 1st condition, our synthesis system was used in its usual mode, concatenating and transforming segments from the synthesis database. For the 2nd condition, a recording of the target lyrics, sung on a flat pitch and with a regular rhythm by the same singers than our databases (RT and MS), was used for the synthesis, instead of the usual synthesis databases RT and MS, to get rid of the units selection and concatenation process. Then 2 different evaluations were conducted. Some examples of sounds synthesized for this evaluation are given in sounds 7.4 to 7.7.
The 1st one consisted in a standard MOS evaluation, were all systems were compared to natural singing from the same songs extracts by the same singers, and to 3 degraded conditions where the natural singing was transformed using an autotune, an overdrive and a time-stretching effect, as described in [Feu+16]. In this evaluation, both the SVP and PaN engines were rated similarly, below the natural voice, but better than the other system and all degraded conditions. Both synthesis conditions, with and without the concatenation were rated similarly in this test, which suggests that the concatenation process does not introduce too much artifacts in the synthesis.
The 2nd evaluation consisted in paired comparisons of short extracts produced by the different synthesis systems, to evaluate both the quality of the articulation and of the expression ("melodic") modeling. In this evaluation, the synthesis of the PaN and SVP engines were rated better than that of the Calliphony system. The 2nd condition without the units concatenation process was prefered to the 1st condition using the database in this test, which suggests that their is nevertheless still some improvement possible on the unit selection and concatenation process to increase the quality of the synthesis. Finally, the PaN engine was also rated slightly better than the SVP engine, although this last result is not very significant.
Compared to the simple evaluation procedure from the Interspeech challenge, such an evaluation, imposing similar conditions for all systems and using real singing extracts from the same singers than the synthesis database, already provides a

more accurate assessment of the systems' qualities, along with some interesting insights on the causes of the limited quality and on the possible improvements. All the details related to this evaluation are given in [Feu+16].
Regarding our system, it would also be interesting to further compare the SVP and PaN engine on specific aspects like the quality of the transposition.

Besides subjective evaluations, objective evaluations may be used to evaluate certain aspects that do not depend on perception, like the efficiency of the proposed algorithm (computation cost and time). In order to alleviate the need of conducting numerous time-consuming listening tests to assess every aspects of singing synthesis, it would be beneficial if one may also use such objective measures to assess the quality of the synthesis or the expression control. For assessing audio and speech quality, some objective measures, that are assumed to correlate with perception, have been proposed [CJG09; Bee+13]. But if such measures may be well suited to identify various distortions related to the coding and transmission of speech in telecommunication systems, one may doubt of their applicability in the case of singing. In [Umb15], the authors evoked the possibility to establish such measures to provide ratings for particular features such as timing, vibrato, tuning, voice quality, or the overall performance expression, independently of singing style. But the development of such measures would be complex and the results obtained would be questionable, considering that singing is also subject to aesthetic considerations with cultural influences.
A proper objective evaluation of a synthesis would ideally require to compare it to some ground truth data (e.g. an audio file, an f_0 curve, ...). But singing is not a deterministic process, and due to the high flexibility of human voice, with infinite possibilities of different timbres and expressions, such ground truth may often not exist, which is fortunately what provides to singing voice, and more generally music, all its richness.

Appendix A

Sampa phonetic alphabet

SAMPA	EXAMPLES
i	idiot, ami
e	ému, été
E	perdu, maison
a	alarme, patte
O	obstacle, corps
o	auditeur, beau
u	coupable, loup
y	punir, élu
2	creuser, deux
9	malheureux, peur
@	petite, fortement
e~	peinture, matin
a~	vantardise, temps
o~	rondeur, bon
9~	lundi, brun
j	piétiner, choyer
w	quoi, fouine
H	lui, bruit
p	patte, repas, cap
t	tête, net
k	carte, écaille, bec
b	bête, habile, robe
d	dire, rondeur, chaude
g	gauche, égal, bague
f	feu, affiche, chef
s	soeur, assez, passe
S	chanter, machine, poche
v	vent, inventer, rêve
z	zéro, raisonner, rose
Z	jardin, manger, piège
l	long, élire, bal
R	rond, charriot, sentir
m	madame, aimer, pomme
n	nous, punir, bonne
N	parking, ping-pong
_	silence marker

FIGURE A.1: List of French SAMPA characters used in this thesis

Appendix B

The CART algorithm

In section 5.4, we explained how we used decision trees to build style models and choose appropriate control parameters for synthesis according to musical contexts. We used for this purpose an implementation of the CART algorithm (for "Classification And Regression Tree") [Bre+84]. For the sake of completeness, we detail here the mechanism behind this algorithm.

The aim of the CART algorithm is to build a tree that allows to predict some class or features based on contextual factors, by learning some simple decision rules from a given dataset associating the features to be predicted to contextual factors. This is done by sequentially splitting the dataset at each step into two new smaller subsets, on the basis of binary questions about the context. For this purpose, the algorithm relies on a "goodness-of-split" evaluation function, commonly called "impurity" function, related to the homogeneity (similarity) of the features contained in each generated subset, and some stopping criteria. In regression trees, the mean-square error (MSE) is commonly used as an impurity function. At each step of the algorithm, for each of the current terminal nodes, the best question that minimizes the total impurity (MSE) on the resulting subnodes is chosen, in a locally optimal fashion. We summarize below the procedure for building a tree.

Let Q denote a set of binary questions about the contextual factors, n a node in the tree, $D(n) = (X, Y)$ the data contained at node n where X are the contextual factors and Y the features to be modeled, and $G(q, n)$ the total impurity of the 2 child nodes obtained when asking the question $q \in Q$ at node n. In the following, we call a "tested node" a node for which we have already evaluated $G(q, n)$ for all questions $q \in Q$ and either split the node or designated it as a terminal node. Then, the steps defined in algorithm 1 below are followed.

Algorithm 1 Decision tree construction (CART algorithm)

1. **Start** with all samples at the root node
2. **While** there are untested nodes **do**
   2.1. Select some untested node n
   2.2 Evaluate $G(q, n)$ for all possible questions $q \in Q$ at this node.
   2.3 Select for this node the question $\hat{q}$ that minimizes the function $G(q, n)$:
   $\hat{q} = \text{argmin}_Q(G(q, n))$
   2.3.1 **If** a stopping criterion is met, declare this node as terminal.
   2.3.2. **else** create two new child nodes. All samples that answer positively to the question are transferred to the left child node n_{q+} and all other samples are transferred to the right child node n_{q-}.

In our approach, the total impurity function $G(n, q)$ is defined by equation B.1 below:

$$c_n = \frac{1}{N_n} \sum_{i=1}^{N_n} y_i$$

$$H(n) = \frac{1}{N_n} \sum_{i=1}^{N_n} (y_i - c_n)^2 \tag{B.1}$$

$$G(q, n) = \frac{N_{n_{q+}}}{N_n} H(n_{q+}) + \frac{N_{n_{q-}}}{N_n} H(n_{q-})$$

where y_i is the value of feature y for the i$^{\text{th}}$ sample, $q+$ represents a positive answer and $q-$ a negative answer to question q, $H(n)$ is the impurity at node n (here the MSE), and N_n is the number of samples present at node n.
A possible stopping criterion that we used is to keep a minimum number, or percentage, of samples on each leaf (terminal node) of the tree.

As some of the contextual factors used in our system are continuous numerical values, we have $q = (j, t_{j,n})$ consisting of a contextual factor j and a threshold $t_{j,n}$. The optimal value for $t_{j,n}$ can be chosen among a set of discrete points based on the available data. Boolean contexts can then be considered similarly, simply by setting $t_{j,n} = 0.5$.

For multi-target decision-trees [Bor+15], where several correlated features are tied together in a single tree as we use in our approach for f_0 and intensity segments, the same steps as for the basic CART algorithm are followed, the only difference being the redefinition of the impurity measure of a node as the sum of squared errors over all features.

Appendix C

Lists of contextual factors

In this section, we list all the contextual factors that have been used for building the styles models for generating the phonemes durations, f_0 segments' parameters and intensity parameters, as described in chapter 5. Note that in the following, a "musical phrase" is defined as the group of notes comprised between 2 silences.

C.1 for phonemes durations models

- consPos: position of the consonant inside the cluster, in case there are several successive consonants (0 is 1st one, 1 is middle one, and 2 is last one)
- currentNoteLenSec: current note duration in seconds
- currentNotePitch: pitch of current note (in midi value)
- hasSeveralCons: there are several clustered consonants in the note
- isAscendingScale: the current note is in an ascending scale (previous note lower and next note higher)
- isDescendingScale: the current note is in a descending scale (previous note higher and next note lower)
- isFirstCons: 1 if the consonant is the first of a cluster, 0 otherwise, or -1 if the consonant is not in a cluster
- isMidCons: 1 if the consonant is in the middle of a cluster, 0 otherwise, or -1 if the consonant is not in a cluster
- isLastCons: 1 if the consonant is the last of a cluster, 0 otherwise, or -1 if the consonant is not in a cluster
- isFirstNote: the current note is the first note of a musical phrase
- isLastNote: the current note is the last note of a musical phrase
- isHighestNote: the current note is the highest note of a musical phrase
- isLowestNote: the current note is the lowest note of a musical phrase
- isMelodicPeak: the current note is a melodic peak (previous and next notes are lower)
- isMelodicValley: the current note is a melodic valley (previous and next notes are higher)
- isPenultimateNote: the current note is the penultimate of a musical phrase
- isSilence: the current note is a silence (if the consonants are on the attack of the first note of a musical phrase)
- nextIntervalJoint: the interval with next note is inferior to 1 tone
- nextNoteDiffLenSec: difference of duration with next note (in seconds)
- nextNoteIsCaduque: the next note is caduc (= next vowel is either /9/ or /@/ and is last note)
- nextNoteIsHighest: the next note is the highest of the musical phrase

- nextNoteIsLowest: the next note is the lowest of the musical phrase
- nextNoteIsMelodicPeak: the next note is a melodic peak (higher than its 2 surrounding notes)
- nextNoteIsMelodicValley: the next note is a melodic valley (lower than its 2 surrounding notes)
- nextNoteLenSec: duration of next note (in seconds)
- nextNotePitch: pitch of next note (in midi value)
- nextNotePitchDiff: pitch difference with next note (in number of semitones)
- nextNotePitchEqual: the current and next notes have the same pitches
- nextNotePitchHigher: the next note has a higher pitch
- nextNotePitchLower: the next note has a lower pitch
- nextNoteSameDur: the next note has a similar duration to the current one (according to a given threshold relative to the tempo)
- nextNoteLonger: next note is longer than the current one (according to a given threshold relative to the tempo)
- nextNoteShorter: next note is shorter than the current one (according to a given threshold relative to the tempo)
- nextPhonIsNASAL: the next phoneme is a nasal consonant
- nextPhonIs[SEMI_VOWEL; SILENCE; sUNVOICED_FRICATIVE; UNVOICED_PLOSIVE; VOICED_PLOSIVE; VOICED_FRICATIVE; VOWEL]: the next phoneme is a [semi-vowel; silence; unvoiced fricative; unvoiced plosive; voiced plosive; voiced fricative; vowel]
- nextPhonIs_/x/: the next phoneme is a /x/ (where /x/ should be replaced by any consonant)
- prevPhonIs[SEMI_VOWE; SILENCE; UNVOICED_FRICATIVE; UNVOICED_PLOSIVE; VOICED_FRICATIVE; VOICED_PLOSIVE; VOWEL]: the previous phoneme is a [semi-vowel; silence; unvoiced fricative; unvoiced plosive; voiced plosive; voiced fricative; vowel]
- prevPhonIs_/x/: the previous phoneme is a /x/ (where /x/ should be replaced by any consonant)
- noteNbCons: number of succeeding consonants in current note
- prevConsLen: duration of preceding consonant if any (otherwise = -1)
- prevNoteIsHighest: previous note is the highest of the musical phrase
- prevNoteIsLowest: previous note is the lowest of the musical phrase
- prevNoteLenSec: duration of the previous note is seconds
- prevNotePitch: pitch of the previous note (in midi value)

C.2 for f_0 models

C.2.1 for sustain segments

- currentNoteLenSec: absolute note duration (seconds)
- currentNotePitch: note pitch as midi value
- isFirstNote: is first note of a musical phrase (= preceded by a silence)
- isPenultimateNote: is penultimate note of the musical phrase (= next note is the last one)
- isLastNote: is last note of the musical phrase (= followed by a silence)

- noteIsCaduque: note is the last note of a musical phrase and vowel is /9/ or /@/
- isLowestNote: is lowest note of the musical phrase
- isHighestNote: is highest note of the musical phrase
- isMelodicPeak: is a melodic peak (= left and right notes have lower pitches)
- isMelodicValley: is a melodic valley (= left and right notes have higher pitches)
- nextIntervalJoint: the pitch interval with next note is <= 2 semitones
- nextNoteDiffLenSec: difference of duration with next note (in seconds)
- nextNoteLenSec: next note duration (in seconds)
- nextNoteLonger: next note is longer than current one (according to a given threshold relative to the tempo)
- nextNotePitchDiff: pitch difference with next note in semitones
- nextNotePitchEqual: next note has same pitch than current one
- nextNotePitchHigher: next note has a higher pitch
- nextNotePitchLower: next note has a lower pitch
- nextNoteSameDur: next note has a similar duration than current note (according to a given threshold relative to the tempo)
- nextNoteShorter: next note is shorter (according to a given threshold relative to the tempo)
- prevIntervalJoint: the pitch interval with the previous note note is <= 2 semitones
- prevNoteDiffLenSec: difference of duration with next note (in seconds)
- prevNoteLenSec: duration of previous note (in seconds)
- prevNoteLonger: previous note is longer than the current one (according to a given threshold relative to the tempo)
- prevNotePitchDiff: pitch difference with previous note in semitones
- prevNotePitchEqual: previous note has the same pitch
- prevNotePitchHigher: previous note has a higher pitch
- prevNotePitchLower: previous note has a lower pitch
- prevNoteSameDur: previous note has a similar duration (according to a given threshold relative to the tempo)
- prevNoteShorter: previous note is shorter (according to a given threshold relative to the tempo)
- vowelLen: duration of the vowel in s

C.2.2 for transition segments

- isFirstNote: left note is the first note of the musical phrase
- isMelodicPeak: the left note of the transition is a melodic peak
- isMelodicValley: the left note of the transition is a melodic valley
- isPenultimateNote: the left note of the transition is the penultimate note of the musical phrase
- jointInterval: the interval of the transition is <= 1 tone
- leftNoteIsHighest: the left note of the transition is the highest note of the musical phrase
- leftNoteIsLowest: the left note of the transition is the lowest note of the musical phrase

- leftNoteLenSec: duration of the left note (in second)
- leftNotePitch: pitch of the left note (in midi value)
- notesLenSecDiff: difference of duration between left and right notes (in seconds)
- notesSameDur: the 2 notes around the transition have a similar durations (according to a given threshold relative to the tempo)
- pitchDiff: pitch difference between the left and right notes (in semitones)
- rightNoteIsCaduque: the right note of the transition is "caduque" (last note and phoneme is /@/ or /9/)
- rightNoteIsHighest: the right note is the highest of the musical phrase
- rightNoteIsLowest: the right note is the lowest of the musical phrase
- rightNoteLenSec: duration of the right note (in seconds)
- rightNoteLonger: the right note is longer than the left one (according to a given threshold relative to the tempo)
- rightNotePitch: pitch of the right note (in midi value)
- rightNoteShorter: the right note is shorter than the left one (according to a given threshold relative to the tempo)
- prevNoteSameDur: previous note has a similar duration (according to a given threshold relative to the tempo)
- prevNoteShorter: previous note is shorter (according to a given threshold relative to the tempo)
- hascons: transition contains consonants (between the 2 vowels of the surrounding notes)
- totalConsLenSec: total cumulated duration of the consonants in the transition (between the 2 vowels)
- hasSemiVowel: there is a semi-vowel in the transition
- hasVoicedCons: there is a voiced consonant in the transition
- hasVoicedPlosive: there is a voiced plosive in the transition
- hasVoicedFricative: there is a voiced fricative in the transition
- hasVoicedFricativeOrPlosive: there is either a voiced plosive or a voiced fricative (or both) in the transition
- hasNasal: there is a nasal in the transition
- hasL, hasR, has_b, has_d, has_g, has_v, has_z, has_Z: there is a /l/ (resp. /R/, /b/, /d/, /g/, /v/, /z/, /Z/) in the transition

C.2.3 for attack and release segments

- hascons: there is one or several consonants at the attack (resp. release) of the note
- hasVoicedCons: note is attacked (resp. released) with one or several voiced cons
- hasSemiVowel: there is a semi-vowel at the beginning of the attacked note (resp. the end of the released note)
- has[VoicedPlosive; VoicedFricative; VoicedfricativeOrPlosive; Unvoiced-Plosive; UnvoicedFricative; UnvoicedFricativeOrPlosive; Nasal]: there is a [voiced plosive; voiced fricative; voiced fricative or voiced plosive; unvoiced plosive; unvoiced fricative; unvoiced fricative or unvoiced plosive; nasal] at the beginning of the attacked note (resp. the end of the released note)

- has_b, has_d, has_g, has_v, has_z, has_Z, hasL, hasR: there is a /b/ (resp. /d/, /g/, /v/, /z/, /Z/) at the start of the attacked (resp. end of the released) note.
- noteLenSec: duration of the attacked (resp. released) note (in seconds)
- currentNotePitch: pitch of the attacked (resp. released) note (in midi value)
- totalConsLenSec: total cumulated duration of the consonants at the start of the attacked (resp. the end of the released) note (in seconds)
- vowelLen: duration of the vowel of the attacked (resp. released) note (in seconds)

C.3 for intensity models

- currentNoteLenSec: duration of current note (in seconds)
- currentNotePitch: pitch of current note (in midi value)
- isAscendingScale: the current note is in an ascending scale (previous note lower and next note higher)
- isDescendingScale: the current note is in a descending scale (previous note higher and next note lower)
- isFirstNote: the current note is the first note of a musical phrase (= preceded by a silence)
- isSecondNote: the current note is the second note of a musical phrase
- isPenultimateNote: the current note is penultimate note of the musical phrase (= next note is the last one)
- isLastNote: the current note is last note of the musical phrase (= followed by a silence)
- noteIsCaduque: the current note is the last note of a musical phrase and the vowel is /9/ or /@/
- nextNoteIsCaduque: the next note is caduque
- isHighestNote: the current note is the highest note of the musical phrase
- isLowestNote: the current note is the lowest note of the musical phrase
- isMelodicPeak: the current note is a melodic peak (= left and right notes have lower pitches)
- isMelodicValley: the current note is a melodic valley (= left and right notes have higher pitches)
- nextNoteIsHighest: the next note is the highest of the musical phrase
- nextNoteIsLowest: the next note is the lowest of the musical phrase
- nextNoteLenSec: duration of the next note (in seconds)
- nextNoteLonger: the next note is longer than the current one (according to a given threshold relative to the tempo)
- nextNotePitch: pitch of the next note (in midi value)
- nextNotePitchDiff: pitch difference with next note in semitones
- nextNotePitchEqual: the next note has the same pitch than current note
- nextNotePitchHigher: the next note has a higher pitch than the current one
- nextNotePitchLower: the next note has a lower pitch than the current one
- nextNoteSameLen: the next note has a similar duration than the current one (according to a given threshold relative to the tempo)

- nextNoteShorter: the next note is shorter than the current one (according to a given threshold relative to the tempo)
- prevNoteIsHighest: the previous note is the highest of the musical phrase
- prevNoteIsLowest: the previous note is the lowest of the musical phrase
- prevNoteLenSec: duration of the previous note (in seconds)
- prevNotePitch: pitch of the previous note (in midi value)
- posInSentence: position of the note in the musical phrase (between 0 and 1, 0 being for the 1st note et 1 for the last one)
- sentenceLenSec: total duration of the musical phrase in seconds
- sentenceLoudness: target normalized mean loudness value for the musical phrase
- sentenceNbNotes: number of notes in the musical phrase
- vowelLenSec: duration of the vowel (in seconds)

The previous contextual factors are used for predicting both the static and dynamic loudness parameter. In addition, the 2 following factors are used to predict the dynamic parameters:

- noteLoudness: target static loudness value for the current note
- deltaLoudness: difference of target static loudness values between the current and previous notes

Appendix D

List of publications

International conferences:

- Ardaillon, L., Degottex, G., & Roebel, A. (2015, September). A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls. In Interspeech 2015.
- Degottex, G., Ardaillon, L., & Roebel, A. (2016, March). Simple multi frame analysis methods for estimation of amplitude spectral envelope estimation in singing voice. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016 (pp. 4975-4979).
- Ardaillon, L., Chabot-Canet, C., & Roebel, A. (2016, September). Expressive control of singing voice synthesis using musical contexts and a parametric F0 model. In Interspeech 2016 (Vol. 2016, pp. 1250-1254).
- Feugère, L., d'Alessandro, C., Delalez, S., Ardaillon, L., & Roebel, A. (2016, September). Evaluation of Singing Synthesis: Methodology and Case Study with Concatenative and Performative Systems. In Interspeech 2016 (pp. 1245-1249).
- Ardaillon, L. & Roebel, A. (2017, September). "A mouth opening effect based on pole modification for expressive singing voice transformation". In Interspeech 2017.

National conferences:

- Ardaillon, L., Roebel, A., & Chabot-Canet, C. (2016, April). Modélisation des paramètres de contrôle pour la synthèse de voix chantée. In CFA/VISHNO 2016.

Contribution to a journal paper:

- Gilles Degottex, Luc Ardaillon, & Axel Roebel. "Multi-Frame Amplitude Envelope Estimation for Modification of Singing Voice". In: IEEE/ACM Transactions on Audio Speech and Language Processing 24.7 (2016), pp. 1242–1254.

Seminars, workshops:

- Chabot-Canet, C., Ardaillon, L., & Roebel, A. (2017, proceedings waiting for publication). Analyse du style vocal et modélisation pour la synthèse de chant expressif: l'exemple d'Edith Piaf. In colloque international "La voix dans les chansons: approches musicologiques", Lyon, 03/03/2016.
- Ardaillon, L., & Roebel, A. (2014, July). Synthèse concaténative de la voix chantée. In Journées des Jeunes Chercheurs en Audition, Acoustique musicale et Signal (JJCAAS), 2014.
- Ardaillon, L., & Roebel, A. (2016, November). A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls. In JJCAAS 2016.
- Summer school "Sciences et Voix : expressions, usages et prises en charge de l'instrument vocal humain", 26-30/09/2016

Master's thesis (supervision):

- Dickerson, M. (2016, July). Modification expressive de la voix chantée. IRCAM.

Undergraduate internship's report (supervision):

- Sébal, L. (2014, may). Stage sur le projet ChaNTeR. Enregistrement des chanteurs et traitement des bases de données.

Appendix E

List of audio files

All sounds referenced here can be accessed from the following url:
http://recherche.ircam.fr/anasyn/ardaillon/these/these.php

Chapter 3:

- 3.1 Soud example from the RT database for word "coïnculpé" (/ _ k O e k y l p e _ /)
- 3.2 Soud example from the RT database for word "ovni" (/ _ O v n i _ /)
- 3.3 Soud example from the MS database for word "ovni" (/ _ O v n i _ /)
- 3.4 Soud example from the MS database for word "myosotis" (/ _ m j O z O t i s _ /)
- 3.5 Soud example from the EL database for word "myosotis" (/ _ m j O z O t i s _ /)
- 3.6 Soud example from the EL database for words "parking lapin" (/ _ p a R k i N l a p e _ /)
- 3.7 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the SVP engine and the RT database
- 3.8 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the SVP engine and the MS database
- 3.9 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the SVP engine and the EL database
- 3.10 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the PaN engine and the RT database
- 3.11 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the PaN engine and the MS database
- 3.12 Copy synthesis (f_0 and durations extracted from a real recording of "Les feuilles mortes" by RT) using the PaN engine and the EL database

Chapter 4:

- 4.1 Synthesis using an upward release (positive value for depth parameter of release segment)
- 4.2 Synthesis with transition from figure 4.9 a).
- 4.3 Synthesis with transition from figure 4.9 b).
- 4.4 Synthesis with transition from figure 4.9 c).

- 4.5 Synthesis with transition from figure 4.9 d).
- 4.6 Synthesis with transition from figure 4.9 e).
- 4.7 Synthesis with transition from figure 4.9 f).

Chapter 5:

- 5.1 Copy synthesis (f_0 and durations extracted from a real recording of "Carmen" from Bizet) using the SVP engine and the RT database with pop/variety timbre style.
- 5.2 Copy synthesis (f_0 and durations extracted from a real recording of "Carmen" from Bizet) using the SVP engine and a restricted database from RT with a lyrical timbre style.
- 5.3 Resynthesis of the voice from an extract from "Les feuilles mortes" by Edith Piaf based on an harmonic partial analysis conducted on the original commercial recording with musical accompaniment.
- 5.4 Synthesis of the f_0 curve extracted from the original recording of "Les feuilles mortes" by Edith Piaf, using a single sinusoid.
- 5.5 Synthesis of the f_0 curve generated by the proposed f_0 model for the original recording of "Les feuilles mortes" by Edith Piaf, using a single sinusoid, after extracting the model parameters from the original f_0 curve.

Chapter 6:

- 6.1 Example of morphing-based intensity timbre transformation
- 6.2 Synthesis of vowels without applying vowel-dependant loudness correction factors.
- 6.3 Synthesis of vowels using vowel-dependant loudness correction factors.
- 6.4 Original recording of loud "clean" voice.
- 6.5 Amplitude modulation applied on sound 6.4 with a sinusoid at $f_0/2$ as a modulating signal.
- 6.6 Isolated sub-harmonics (difference between sounds 6.4 and 6.5).
- 6.7 High-pass filtered sub-harmonics.
- 6.8 Synthetic rough voice obtained by mixing sound 6.4 with sound 6.7.
- 6.9 2^{nd} example of amplitude modulation applied on sound 6.4, using a sum of 3 sinusoids at $f_0/2$, $f_0/3$, and $f_0/6$ as a modulating signal.
- 6.10 Isolated sub-harmonics from sound 6.9.
- 6.11 High-pass-filtered sub-harmonics from sound 6.10.
- 6.12 Synthetic rough voice obtained by mixing sound 6.4 with sound 6.11.
- 6.13 Original recording with a roughness (growl) effect by MS singer.
- 6.14 Original recording of same extract than sound 6.13 but without roughness.
- 6.15 Synthetic roughness (growl) effect applied on sound 6.14.

- 6.16 Original recording of a rough (shouted) voice.
- 6.17 Synthesis of shouted voice from sound 6.16 using PaN without roughness (no jitter/shimmer).
- 6.18 Synthesis of shouted voice from sound 6.16 using PaN with original jitter and shimmer.
- 6.19 Original recording of a rough (shouted) voice by MS singer.
- 6.20 Synthesis of shouted voice from sound 6.19 using PaN without roughness (no jitter/shimmer).
- 6.21 Synthesis of shouted voice from sound 6.19 using PaN with original jitter scaled with a factor 0.5.
- 6.22 Synthesis of shouted voice from sound 6.19 using PaN with original jitter scaled with a factor 2.
- 6.23 Original recording of "clean" loud voice without roughness by MS singer.
- 6.24 Jitter and shimmer extracted from sound 6.19 applied on sound 6.23 using PaN.

Chapter 7:

- 7.1 Extract of synthesis for the opera I.D. by Arnaud Petit with the EL database (with midi musical accompaniment).
- 7.2 A capella version of the song "Les feuilles d'Interspeech" submitted to the singing synthesis challenge at the Interspeech 2017 conference, using the PaN engine, RT database, and Le Roux style model.
- 7.3 Sound 7.2 with musical accompaniment.
- 7.4 Synthesis of song "les feuilles d'Interspeech" with the PaN engine and the MS database, used for the evaluation in [Feu+16].
- 7.5 Synthesis of song "les feuilles d'Interspeech" with the PaN engine and the MS database, used for the evaluation in [Feu+16].
- 7.6 Synthesis of the song "Au temps d'Interspeech" with the SVP engine and the RT database, used for the evaluation in [Feu+16].
- 7.7 Synthesis of the song "Au temps d'Interspeech" with the PaN engine and the MS database, used for the evaluation in [Feu+16].

Bibliography

[ABS09] Vipul Arorat, Laxmidhar Behera, and Pradip Sircar. “Singing Voice Synthesis For Indian Classical Raga System”. In: *Signals and Systems Conference (ISSC)*. 2009.

[ACR16a] Luc Ardaillon, C Chabot-canet, and Axel Roebel. “Modélisation des paramètres de contrôle pour la synthèse de voix chantée”. In: *CFA / VISHNO 2016*. 2016, pp. 2241–2247.

[ACR16b] Luc Ardaillon, Celine Chabot-Canet, and Axel Roebel. “Expressive control of singing voice synthesis using musical contexts and a parametric F0 model”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 08-12-Sept. 2016, pp. 1250–1254.

[AD03] Christophe Alessandro and Boris Doval. “Voice quality modification for emotional speech synthesis.” In: *Eighth European Conference on Speech Communication and Technology.* 2003.

[ADC98] Christophe Alessandro, Boris Doval, and Orsay Cedex. “Experiments in voice quality modification of natural speech signals: The spectral approach”. In: *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis.* (1998).

[ADR15] Luc Ardaillon, Gilles Degottex, and Axel Roebel. “A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2015, pp. 3375–3379.

[AH71] B. S. Atal and S. L. Hanauer. “Speech Analysis and Synthesis by Linear Prediction of the Speech Wave”. In: *The Journal of the Acoustical Society of America* 50.April (1971), pp. 637–655.

[AK00] Masato Akagi and Hironori Kitakaze. “Perception of synthesized singing voices with fine fluctuation in their fundamental frequency contours”. In: *Sixth International Conference on Spoken Language Processing (ICSLP)*. 2000.

[Alk92] Paavo Alku. “Glottal wave analysis with Pitch Synchronous Iterative Adaptive Inverse Filtering”. In: *Speech communication* 11.2-3 (1992), pp. 109–118.

[Alo04] Marcos Alonso. “Model d’Expressivitat Emocional per a un Sintetitzador de Veu Cantada”. Master thesis. Universitat Pompeu Fabra, Barcelona, Spain, 2004.

[AR17] Luc Ardaillon and Axel Roebel. “A mouth opening effect based on pole modification for expressive singing voice transformation”. In: *Interspeech*. Stockholm, Sweden, 2017.

[Ard13] Luc Ardaillon. “Synthèse du chant”. Master thesis. Université Paris VI Pierre et Marie Curie (UPMC), Paris, France, 2013.

[Bai09] Lucie Bailly. “Interaction entre cordes vocales et bandes ventriculaires en phonation : exploration in-vivo , modélisation physique , validation”. PhD thesis. Université du Maine, Le Mans, France, 2009.

[Bat04] Bret Battey. “Bézier Spline Modeling of Pitch-Continuous Melodic Expression and Ornamentation”. In: *Computer Music Journal* 28.4 (2004), pp. 25–39.

[BB13] Jordi Bonada and Merlijn Blaauw. “Generation of growl-type voice qualities by spectral morphing”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 6910–6914.

[BB16a] Merlijn Blaauw and Jordi Bonada. *A Singing Synthesizer Based on PixelCNN*. Barcelona, 2016.

[BB16b] Jordi Bonada and Merlijn Blaauw. “Expressive Singing Synthesis based on Unit Selection for the Singing Synthesis Challenge 2016”. In: *INTERSPEECH*. 2016, pp. 1230–1234.

[BB17] Merlijn Blaauw and Jordi Bonada. “A Neural Parametric Singing Synthesizer”. In: *Interspeech*. 2017.

[BBL05] N Barbot, O. Boëffard, and D. Lolive. “F0 stylisation with a free-knot B-spline model and simulated-annealing optimization”. In: *Ninth European Conference on Speech Communication and Technology*. 2005, pp. 325–328.

[Bec01] Frédéric Bechet. “Lia_phon: Un système complet de phonétisation de textes”. In: *Traitement automatique des langues* 42.1 (2001), pp. 47–67.

[Bee+13] John G. Beerends et al. “Perceptual Objective Listening Quality Assessment (POLQA), The Third Generation ITU-T Standard for End-to-End Speech Quality Measurement Part II-Perceptual Model”. In: *Journal of the Audio Engineering Society* 61.6 (2013).

[Bel09] Greg Beller. “Analyse et modèle génératif de l’expressivité : application à la Parole et à l’Interprétation musicale”. PhD thesis. Université Paris VI - Pierre et Marie Curie, Paris, France, 2009.

[Ber+14] Nathalie Henrich Bernardoni et al. “Vocal tract resonances in singing: variation with laryngeal mechanism for male operatic singers in chest and falsetto registers”. In: *The Journal of the Acoustical Society of America* 135.1 (2014), pp. 491–501.

[Ber96] G. Berndtsson. “The KTH rule system for singing synthesis”. In: *Computer Music Journal* 20.1 (1996), pp. 76–91.

[BF99] Roberto Bresin and Anders Friberg. “Synthesis and decoding of emotionally expressive music performance”. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Vol. 4. February 1999. 1999, pp. 317–322.

[Bir07] Peter Birkholz. “Articulatory Synthesis of Singing”. In: *Proceedings of Interspeech*. 2007, pp. 4001–4004.

[Bjo08] Eva Bjo. “Musical Theater and Opera Singing — Why So Different ? A Study of Subglottal Pressure , Voice Source , and Formant Frequency Characteristics”. In: *Journal of Voice* 22.5 (2008), pp. 533–540.

[BK06] Paul Boersma and Gordana Kovacic. "Spectral characteristics of three styles of Croatian folk singing." In: *The Journal of the Acoustical Society of America* 119.3 (2006), pp. 1805–1816.

[BL03] Jordi Bonada and Alex Loscos. *Sample-based singing voice synthesizer by spectral concatenation*. 2003.

[Bla02] Alan W Black. "Perfect synthesis for all the people all of the time". In: *Proceedings of 2002 IEEE Workshop on Speech Synthesis*. 2002, pp. 167–170.

[Bog+04] Niels Bogaards et al. "Sound Analysis and Processing with AudioSculpt 2". In: *International Computer Music Conference (ICMC)*. 2004.

[Boh+91] L. R. Bohl et al. "Decision Trees for Phonological Rules in Continuous Speech". In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1991, pp. 185–188.

[Bon+01a] Jordi Bonada et al. "Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models". In: *ICMC*. 2001.

[Bon+01b] Jordi Bonada et al. "Spectral Approach to the Modeling of the Singing Voice". In: *Audio Engineering Society Convention 111*. 2001.

[Bon+11] Jordi Bonada et al. "Spectral Processing". In: *Digital Audio Effects*. Ed. by U Zölzer. John Wiley & Sons, 2011. Chap. 10, pp. 393–445.

[Bon04] Jordi Bonada. "High quality voice transformations based on modeling radiated voice pulses in frequency domain". In: *Proc. Digital Audio Effects (DAFx)*. 3. 2004, pp. 291–295.

[Bon08a] Jordi Bonada. "Voice Processing and synthesis by performance sampling and spectral models". PhD thesis. Universitat Pompeu Fabra, Barcelona, Spain, 2008, p. 251.

[Bon08b] Jordi Bonada. "Wide-band harmonic sinusoidal modeling". In: *Proc of the 11th Int Conference on Digital Audio Effects (DAFx08)*. 2008.

[Bor+15] Hanen Borchani et al. "A survey on multi-output regression". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.5 (2015), pp. 216–233.

[Bre+84] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.

[BS00] Martine E Bestebreurtje and K Schutte. "Resonance Strategies for the Belting Style : Results of a Single Female Subject Study". In: *Journal of voice* 14.2 (2000), pp. 194–204.

[BS02] J Bretos and Johan Sundberg. "Measurements of vibrato parameters in long sustained crescendo notes as sung by ten sopranos". In: *Journal of Voice* 17.3 (2002), pp. 343–352.

[BS07] Jordi Bonada and Xavier Serra. "Synthesis of the Singing Voice by Performance Sampling and Spectral Models". In: *IEEE signal processing magazine* 24.2 (2007), pp. 67–79.

[BTC01] Alan W. Black, Paul Taylor, and Richard Caley. *The Festival Speech Synthesis System - system documentation*. Tech. rep. University of Edinburgh, 2001.

[Can+00] Pedro Cano et al. "Voice Morphing System for Impersonating in Karaoke Applications". In: *ICMC*. 2000.

[Can+04] Sergio Canazza et al. "Modeling and control of expressiveness in music performance". In: *Proceedings of the IEEE* 92.4 (2004), pp. 686–701.

[CB97] Nick Campbell and Alan W Black. "Prosody and the selection of source units for concatenative synthesis". In: *Progress in speech synthesis*. Springer, 1997, pp. 279–292.

[CCM01] Marine Campedel-Oudot, Olivier Cappé, and Eric Moulines. "Estimation of the spectral envelope of voiced sounds using a penalized likelihood approach". In: *IEEE Transactions on Speech and Audio Processing* 9.5 (2001), pp. 469–481.

[CH08] Arturo Camacho and John G. Harris. "A sawtooth waveform inspired pitch estimator for speech and music". In: *The Journal of the Acoustical Society of America* 124.3 (2008), pp. 1638–1652.

[Cha08] Céline Chabot-canet. "Les feuilles mortes ou les avatars d'une chanson culte : aborder les phénomènes vocaux interprétatifs dans la chanson française à travers la pratique de la reprise". In: *L'Éducation musicale* 557/558 (2008), pp. 28–33.

[Cha13] Céline Chabot-Canet. "Interprétation, phrasé et rhétorique vocale dans la chanson française depuis 1950 : expliciter l'indicible de la voix". PhD thesis. Université Lyon II – Louis Lumière, Lyon, France, 2013.

[CJG09] Dermot Campbell, Edward Jones, and Martin Glavin. "Audio quality assessment techniques - A review, and recent developments". In: *Signal Processing* 89.8 (2009), pp. 1489–1500.

[CK02] Alain de Cheveigné and Hideki Kawahara. "YIN, a fundamental frequency estimator for speech and music." In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1917–1930.

[CM96] Olivier Cappe and Eric Moulines. "regularization techniques for discrete cepstrum estimation". In: *IEEE Signal Processing Letters* 3.4 (1996), pp. 100–102.

[Coo05] Perry Cook. "Real-Time Performance Controllers for Synthesized Singing". In: *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. 2005, pp. 236–237.

[Coo89] Perry R Cook. *Synthesis of the singing voice using a physically parameterized model of the human vocal tract*. Tech. rep. CCRMA, Department of music, Stanford University, 1989.

[Coo93] Perry Cook. "SPASM, a real-time vocal tract physical model controller; and singer, the companion software synthesis system". In: *Computer Music Journal* 17.1 (1993), pp. 30–44.

[Coo98] Perry R Cook. "Toward the Perfect Audio Morph ? Singing Voice Synthesis and Processing". In: *Proceedings of the 1st. International Conference on Digital Audio Effects (DAFX)*. Barcelona, 1998.

[Cro80] R. E. Crochiere. "A Weighted Overlap-Add Method of Short-time Fourier Analysis/Synthesis". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.1 (1980), pp. 99–102.

[CS92] Gunilla Carlsson and Johan Sundberg. “Formant Frequency Tuning in Singing”. In: *Journal of Voice* 6.3 (1992), pp. 256–260.

[DAH03] Boris Doval, Christophe Alessandro, and Nathalie Henrich. “The voice source as a causal / anticausal linear filter”. In: *ISCA Tutorial and Research Workshop on Voice Quality: Functions, Analysis and Synthesis*. 2003.

[DAH06] Boris Doval, Christophe Alessandro, and Nathalie Henrich. “The Spectrum of Glottal Flow Models”. In: *Acta acustica united with acustica* 92.6 (2006), pp. 1026–1046.

[DAl94] Chistophe D’Alessandro. “The Pitch of Short-duration Vibrato Tones”. In: 95.3 (1994), pp. 1617–1630.

[DAR16a] Gilles Degottex, Luc Ardaillon, and Axel Roebel. “Multi-Frame Amplitude Envelope Estimation for Modification of Singing Voice”. In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 24.7 (2016), pp. 1242–1254.

[DAR16b] Gilles Degottex, Luc Ardaillon, and Axel Roebel. “Simple multi frame analysis methods for estimation of amplitude spectral envelope estimation in singing voice”. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2016, pp. 4975–4979.

[DD98] RB Dannenberg and Istvan Derenyi. “Combining instrument and performance models for high-quality music synthesis”. In: *Journal of New Music Research* 27.3 (1998), pp. 211–238.

[Deg10] Gilles Degottex. “Glottal source and vocal-tract separation”. PhD thesis. Université Paris VI - Pierre et Marie Curie, Paris, France, 2010, p. 181.

[Deg15] Gilles Degottex. “A time regularization technique for discrete spectral envelopes through frequency derivative”. In: *IEEE Signal Processing Letters* 22.7 (2015), pp. 978–982.

[Dev+11] Johanna C Devaney et al. “Characterizing singing voice fundamental frequency trajectories”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2011, pp. 73–76.

[DGR95] Ph Depalle, G Garcia, and Xavier Rodet. “The recreation of a castrato voice, Farinelli’s voice”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 1995, pp. 242–245.

[DH97] J W Dang and K Honda. “Acoustic characteristics of the piriform fossa in models and humans”. In: *Journal of the acoustical society of America* 101.1 (1997), pp. 456–465.

[Dic16] Maxime Dickerson. “Modification expressive de voix chantée”. Master thesis. Université Paris VI Pierre et Marie Curie (UPMC), Paris, France, 2016.

[DL93] Thierry Dutoit and H. Leich. “MBR-PSOLA: Text-To-Speech synthesis based on an MBE re-synthesis of the segments database”. In: *Speech Communication* 13.3-4 (1993), pp. 435–440.

[DM80] Steven Davis and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 357–366.

[Don+11] Minghui Dong et al. "Spectral Transformation of Singing Vowels by Dynamic Frequency Warping". In: *APSIPA ASC*. Xi'an, China, 2011.

[DRR11a] Gilles Degottex, Axel Roebel, and Xavier Rodet. "Phase Minimization for Glottal Model Estimation". In: *IEEE Trans. Audio, Speech, and Language Processing* 19.5 (2011), pp. 1080–1090.

[DRR11b] Gilles Degottex, Axel Roebel, and Xavier Rodet. "Pitch transposition and breathiness modification using a glottal source model and its adapted vocal-tract filter". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2011, pp. 5128–5131.

[DS12] Gilles Degottex and Yannis Stylianou. "A Full-Band Adaptive Harmonic Representation of Speech". In: *Thirteenth Annual Conference of the International Speech Communication Association*. 2012.

[DSC13] Calzada Defez, Joan Claudi Socor, and Robert A J Clark. "Parametric model for vocal effort interpolation with Harmonics Plus Noise Models". In: *8th ISCA Speech Synthesis Workshop*. Barcelona, Spain, 2013, pp. 25–30.

[Dut+96] Thierry Dutoit et al. "The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes". In: *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*. 1996, pp. 1393–1396.

[EM91] Amro El-Jaroudi and John Makhoul. "Discrete All-Pole Modeling". In: *IEEE transactions on signal processing* 39.2 (1991), pp. 411–423.

[Fan95] Gunnar Fant. "The LF-model revisited . Transformations and frequency domain analysis". In: *STL-QPSR* 36.2-3 (1995), pp. 119–156.

[Fan97] Gunnar Fant. "The voice source in connected speech". In: *Speech communication* 22.2-3 (1997), pp. 125–139.

[FBS09] Anders Friberg, Roberto Bresin, and Johan Sundberg. "Overview of the KTH rule system for musical performance". In: *Advances in Cognitive Psychology* 2.2 (2009), pp. 145–161.

[Feu+16] Lionel Feugère et al. "Evaluation of singing synthesis: Methodology and case study with concatenative and performative systems". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2016, pp. 1245–1249.

[Feu+17] Lionel Feugère et al. "Cantor Digitalis: chironomic parametric synthesis of singing". In: *EURASIP Journal on Audio, Speech, and Music Processing* (2017).

[Feu13] Lionel Feugère. "Synthèse par règles de la voix chantée contrôlée par le geste et applications musicales". PhD thesis. Université Paris VI - Pierre et Marie Curie, Paris, France, 2013.

[FG66] J. L. Flanagan and R. M. Golden. "Phase Vocoder". In: *Bell Labs Technical Journal* 45.9 (1966), pp. 1493–1509.

[FH84] Hiroya Fujisaki and Keikichi Hirose. “Analysis of voice fundamental frequency contours for declarative sentences of Japanese”. In: *Journal of the Acoustical Society of Japan* 5.4 (1984), pp. 233–242.

[FLL85] Gunnar Fant, J. Liljencrants, and Q. Lin. “A four-parameter model of glottal flow”. In: *STL-QPSR* 26.4 (1985), pp. 1–13.

[FM03] Qiang Fu and Peter Murphy. “Adaptive inverse filtering for high accuracy estimation of the glottal source Adaptive Inverse Filtering for High Accuracy Estimation”. In: *ISCA Tutorial and Research Workshop on Non-Linear Speech Processing*. 2003.

[Fon01] Jaume Ortolà i Font. “Musical and phonetic controls in a singing voice synthesizer”. Master thesis. Polytechnics University of Valencia, 2001.

[Fón83] Ivan Fónagy. *La vive voix: essais de psycho-phonétique*. Payot, 1983.

[For73] Jr. Forney G.D. “The viterbi algorithm”. In: *Proceedings of the IEEE* 61.3 (1973), pp. 302–309.

[Fow79] Carol A. Fowler. “"Perceptual centers" in speech production and perception”. In: *Attention, Perception, & Psychophysics* 25.5 (1979), pp. 375–388.

[FPR11] Mary Florentine, Arthur N. Popper, and Richard R.Fay. *Loudness*. Springer, 2011.

[Fri+00] Anders Friberg et al. “Generating Musical Performances with Director Musices”. In: *Computer Music Journal* 24.3 (2000), pp. 23–29.

[Fri71] Charlotte J Frisbie. “Anthropological and Ethnomusicological Implications of a Comparative Analysis of Bushmen and African Pygmy Music”. In: *Ethnology* 10.3 (1971), pp. 265–290.

[Fri91] Anders Friberg. “Generative Rules for Music Performance: A Formal Description of a Rule System”. In: *Computer Music Journal* 15.2 (1991), pp. 56–71.

[FRR09] Snorre Farner, Axel Röbel, and Xavier Rodet. “Natural transformation of type and nature of the voice for extending vocal repertoire in high-fidelity applications”. In: *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. 2009.

[Fux12] Thibaut Fux. “Vers un système indiquant la distance d’un locuteur par transformation de sa voix”. PhD thesis. Université de Grenoble, France, 2012.

[FZ90] Hugo Fastl and Eberhard Zwicker. *Psychoacoustics: Facts and Models*. Springer Berlin Heidelberg, 1990.

[Gar+10] Maëva Garnier et al. “Vocal tract adjustments in the high soprano range”. In: *The Journal of the Acoustical Society of America* 127.6 (2010), pp. 3771–3780.

[Gio+99] Antoine Giovanni et al. “Nonlinear behavior of vocal fold vibration: the role of coupling between the vocal folds.” In: *Journal of voice : official journal of the Voice Foundation* 13.4 (1999), pp. 465–476.

[Gom+03] Emilia Gomez et al. “Melodic characterization of monophonic recordings for expressive tempo transformations”. In: *Proceedings of Stockholm Music Acoustics Conference (Smac)*. 2003.

[GPW04] Werner Goebl, Elias Pampalk, and Gerhard Widmer. “Exploring expressive performance trajectories: six famous pianists play six Chopin pieces”. In: *Proceedings of the 8th international conference on music perception and cognition*. 2004, pp. 505–509.

[GR90] Thierry Galas and Xavier Rodet. “An improved cepstral method for deconvolution of source-filter systems with discrete spectra: Application to musical sound signals”. In: *ICMC*. 1990.

[GR91] Thierry Galas and Xavier Rodet. “Generalized Discrete Cepstral Analysis for Deconvolution of Source-Filter System with Discrete Spectra”. In: *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics* (1991).

[Gra+88] Patricia Gramming et al. “Relationship between changes in voice pitch and loudness”. In: *Journal of Voice* 2.2 (1988), pp. 118–126.

[HC96] YS Hsiao and DG Childers. “A new approach to formant estimation and modification based on pole interaction”. In: *Thirteenth Asilomar Conference on Signals, Systems and Computers*. 1996, pp. 783–787.

[Hen+06] Nathalie Henrich et al. “Period-doubling occurences in singing : the " bassu " case in traditional Sardinian " A Tenore " singing”. In: *ICVPB*. January. Tokyo, 2006.

[HLS17] Hanna Hallqvist, Filipa M B Lã, and Johan Sundberg. “Soul and Musical Theater : A Comparison of Two Vocal”. In: *Journal of Voice* 31.2 (2017), pp. 229–235.

[HMC89] C. Hamon, E. Mouline, and F. Charpentier. “A diphone synthesis system based on time-domain prosodic modifications of speech”. In: *International Conference on Acoustics, Speech, and Signal Processing (*. 1989, pp. 238–241.

[HR14] Stefan Huber and Axel Roebel. “On the use of voice descriptors for glottal source shape parameter estimation”. In: *Computer Speech and Language* 28.5 (2014), pp. 1170–1194.

[HR15] Stefan Huber and Axel Roebel. “Voice quality transformation using an extended source-filter speech model”. In: *12th Sound and Music Computing Conference (SMC)*. 2015, pp. 69–76.

[HSW11] Nathalie Henrich, John Smith, and Joe Wolfe. “Vocal tract resonances in singing: Strategies used by sopranos, altos, tenors, and baritones”. In: *The Journal of the Acoustical Society of America* 129.2 (2011), pp. 1024–1035.

[Hub+99] Jessica E Huber et al. “Formants of children, women, and men: The effects of vocal intensity variation”. In: *The Journal of the Acoustical Society of America* 106.3 (1999), pp. 1532–1542.

[Hub15] Stefan Huber. “Voice Conversion by modelling and transformation of extended voice characteristics”. PhD thesis. Université Paris VI - Pierre et Marie Curie (UPMC), Paris, France, 2015.

[IIO14a] Yukara Ikemiya, Katsutoshi Itoyama, and Hiroshi G. Okuno. “Transcribing vocal expression from polyphonic music”. In: *ICASSP*. 2014, pp. 3151–3155.

[IIO14b] Yukara Ikemiya, Katsutoshi Itoyama, and Hiroshi G Okuno. “Transferring Vocal Expression of F0 Contour using Singing Voice Synthesizer”. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2014, pp. 250–259.

[ISO03] ISO. *International Standard ISO 226: Normal Equal-Loudness Level Contours*. 2003.

[ITU16] ITU-T. *Recommendation ITU-T-P.800.2 : Mean opinion score interpretation and reporting*. Tech. rep. 2016.

[JBB06] Jordi Janer, Jordi Bonada, and Merlijn Blaauw. “Performance-driven control for sample-based singing voice synthesis”. In: *Proc. of DAFx*. 2006, pp. 41–44.

[Jen99] Kristoffer Jensen. “Envelope model of isolated musical sounds”. In: *Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99)*. Trondheim, 1999.

[Jon+01] T M Jones et al. “Objective assessment of hoarseness by measuring jitter”. In: *Clinical Otolaryngology* 26.1 (2001), pp. 29–32.

[JSW04] Elodie Joliveau, John Smith, and Joe Wolfe. “Vocal tract resonances in singing: The soprano voice”. In: *The Journal of the Acoustical Society of America* 116.4 (2004), pp. 2434–2439.

[Kak+09] Tatsuya Kako et al. “Automatic identification for singing style based on sung melodic contour characterized in phase plane”. In: *ISMIR*. 2009, pp. 393–398.

[Kaw16] Hideki Kawahara. “SparkNG: Interactive MATLAB tools for introduction to speech production, perception and processing fundamentals and application of the aliasing-free L-F model component”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2016, pp. 1180–1181.

[Kaw97] Hideki Kawahara. “Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1997, pp. 1303–1306.

[KAZ16] Hideki Kawahara, Yannis Agiomyrgiannakis, and Heiga Zen. “Using instantaneous frequency and aperiodicity detection to estimate F0 for high-quality speech synthesis”. In: *9th ISCA Speech Synthesis Workshop*. 2016, pp. 221–228.

[Ken12] Hideki Kenmochi. “Singing synthesis as a new musical instrument”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 5385–5388.

[Kin+14] Lin Kin et al. “Visualising Singing Style Under Common Musical Events Using Pitch-Dynamics Trajectories and Modified TRACLUS Clustering Visualising Singing Style Under Common Musical Events Using Pitch-Dynamics”. In: *13th International Conference on Machine Learning and Applications (ICMLA)*. 2014, pp. 237–242.

[Kla80] Dennis H Klatt. “Software for a cascade/parallel formant synthesizer”. In: *the Journal of the Acoustical Society of America* 67.3 (1980), pp. 971–995.

[KM12] Alexis Kirke and Eduardo Reck Miranda. *Guide to computing for expressive music performance*. Springer Science & Business Media, 2012.

[KMD99] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigné. "Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds". In: *Speech Communication* 27.3 (1999), pp. 187–207.

[KO07] Hideki Kenmochi and Hayato Ohshita. "VOCALOID – Commercial singing synthesizer based on sample concatenation". In: *Interspeech*. 2007, pp. 4009–4010.

[Kob02] Malte Kob. "Physical modeling of the singing voice". Master thesis. Rheinisch-Westfälischen Technischen Hochschule Aachen, 2002.

[Kob04] Malte Kob. "Analysis and modelling of overtone singing in the sygyt style". In: *Applied Acoustics* 65.12 SPEC. ISS. (2004), pp. 1249–1259.

[KR86] Peter Kabal and Ravi Ramachandran. "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.6 (1986), pp. 1419–1426.

[KV98] Esther Klabbers and Raymond Veldhuis. "On the reduction of concatenation artefacts in diphone synthesis". In: *ICSLP* 98 (1998), pp. 1983–1986.

[LA08] Javier Latorre and Masami Akamine. "Multilevel parametric-base F0 model for speech synthesis". In: *Ninth Annual Conference of the International Speech Communication Association - Interspeech*. 2008, pp. 2274–2277.

[Lag+16] Aude Lagier et al. "The shouted voice: A pilot study of laryngeal physiology under extreme aerodynamic pressure". In: *Logopedics Phoniatrics Vocology* (2016).

[Lai07] Wen-hsing Lai. "F0 control model for Mandarin singing voice synthesis". In: *Second International Conference on Digital Telecommunications ICDT'07*. 2007.

[Lan+08] Pierre Lanchantin et al. "Automatic Phoneme Segmentation with Relaxed Textual Constraints". In: *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. 2008.

[LB04] Alex Loscos and Jordi Bonada. "Emulating rough and growl voice in spectral domain". In: *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*. Naples, Italy, 2004.

[LB13] Jean Sylvain Liénard and Claude Barras. "Fine-grain voice strength estimation from vowel spectral cues". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. August. 2013, pp. 128–132.

[LBB10] Damien Lolive, Nelly Barbot, and Olivier Boeffard. "B-Spline Model Order Selection With Optimal MDL Criterion Applied to Speech Fundamental Frequency Stylization". In: *IEEE Journal of Selected Topics in Signal Processing* 4.3 (2010), pp. 571–581.

[LD97] J Laroche and M Dolson. “Phase-vocoder: about this phasiness business”. In: *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*. 1997.

[LD99a] Jean Laroche and Mark Dolson. “New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 1999, pp. 91–94.

[LD99b] J S Liénard and M G Di Benedetto. “Effect of vocal effort on spectral properties of vowels”. In: *The Journal of the Acoustical Society of America* 106.1 (1999), pp. 411–22.

[LDL12] S W Lee, Minghui Dong, and Haizhou Li. “A study of F0 modelling and generation with lyrics and shape characterization for singing voice synthesis”. In: *8th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. 2012, pp. 150–154.

[LDR10] Pierre Lanchantin, Gilles Degottex, and Xavier Rodet. “A HMM-based speech synthesis system using a new glottal source and vocal-tract separation method”. In: *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. 2010, pp. 4630–4633.

[Lee+12] S W Lee et al. “Generalized F0 modelling with absolute and relative pitch features for singing voice synthesis”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 429–432.

[Lee+14] S. W. Lee et al. “A comparative study of spectral transformation techniques for singing voice synthesis”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2014.

[Lee05] Matthew E Lee. “Acoustic Models for the Analysis and Synthesis of the Singing Voice”. PhD thesis. Georgia Institute of Technology, 2005.

[Lin+01] Per Åke Lindestad et al. “Voice source characteristics in Mongolian "throat singing" studied with high-speed imaging technique, acoustic spectra, and inverse filtering”. In: *Journal of Voice* 15.1 (2001), pp. 78–85.

[Lol06] Damien Lolive. “Comparing B-Spline and Spline Models for F0 Modelling”. In: *Lecture notes in computer science* 4188 (2006), pp. 423–430.

[LR13] Marco Liuni and Axel Röbel. “Phase vocoder and beyond”. In: *Musica/Tecnologia* 7 (2013), pp. 73–89.

[Mac+97a] Michael W. Macon et al. “A singing voice synthesis system based on sinusoidal modeling”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1997, pp. 435–438.

[Mac+97b] Michael W. Macon et al. “Concatenation-based MIDI-to-Singing Voice Synthesis”. In: *103rd Meeting of the AES*. 1997.

[MAH93] Hideyuki Mizuno, Masanobu Abe, and Tomohisa Hirokawa. "Waveform-based speech synthesis approach with a formant frequency modification". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1993, pp. 195–198.

[Mak75] John Makhoul. "linear Prediction: A Tutorial Review". In: *Proceedings of the IEEE* 63.4 (1975), pp. 561–580.

[Mar81] Stephen Michael Marcus. "Acoustic determinants of perceptual center (P-center) location". In: *Attention, Perception, & Psychophysics& Psychophysics* 30.3 (1981), pp. 247–256.

[MB90] Robert Maher and James Beauchamp. "An Investigation of Vocal Vibrato for Synthesis". In: *Applied Acoustics* 30.2-3 (1990), pp. 219–245.

[MBL06] Oscar Mayor, Jordi Bonada, and Alex Loscos. "The Singing Tutor: Expression Categorization and Segmentation of the Singing Voice". In: *Proceedings of the AES 121st Convention*. 2006.

[MBM06] Esteban Maestre, Jordi Bonada, and Oscar Mayor. "Modeling musical articulation gestures in singing voice performances". In: *Proceedings of the AES 121st Convention.* 2006.

[MC90] Eric Moulines and Francis CHARPENTIER. "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones". In: *Speech communication* 9.5-6 (1990), pp. 453–467.

[ME86] John Makhoul and Amro El-Jaroudi. "Time-scale modification in medium to low rate speech coding". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1986, pp. 1705–1708.

[MG76] John D Markel and Augustine H Jr Gray. *Linear prediction of speech*. Ed. by Sringer-Verlag. Vol. 12. New York: Springer Science & Business Media, 1976.

[ML95] Eric Moulines and Jean Laroche. "Non-parametric techniques for pitch-scale and time-scale modification of speech". In: *Speech communication* 16.2 (1995), pp. 175–205.

[MM08] Robert C Maher and A E S Member. "Control of Synthesized Vibrato during Portamento Musical Pitch Transitions". In: *Journal of the Audio Engineering Society* 56.1 (2008), pp. 18–27.

[Mol+14] Emilio Molina et al. "Parametric model of spectral envelope to synthesize realistic intensity variations in singing voice". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 634–638.

[Mor78] Jorge J Moré. "The Levenberg-Marquardt algorithm: implementation and theory". In: *Numerical analysis*. Springer, 1978, pp. 105–116.

[MQ86] Robert J. McAuley and Thomas F. Quatieri. "Speech Analysis/Synthesis Based on a Sinusoidal Representation". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.4 (1986), pp. 744–754.

[MS90] D G Miller and H K Schutte. "Formant Tuning in a Professional Baritone". In: *Journal of Voice* 4.3 (1990), pp. 231–237.

[Muñ+03] J. Muñoz et al. “Acoustic and Perceptual Indicators of Normal and Pathological Voice”. In: *Folia Phoniatrica et logopaedica* 55.2 (2003), pp. 102–114.

[Nak+14] Kazuhiro Nakamura et al. “HMM-based singing voice synthesis and its application to japanese and english”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 265–269.

[Nak04] Ichiro Nakayama. “Comparative Studies on Vocal Expressions in Japanese Traditional and Western Classical- Style Singing , Using a Common Verse”. In: *Proc. ICA*. 2004, pp. 295–296.

[NG09] Tomoyasu Nakano and Masataka Goto. “VOCALISTENER : A SINGING-TO-SINGING SYNTHESIS SYSTEM”. In: July (2009), pp. 23–25.

[NH02] Rgen Neubauer and Hanspeter Herzel. “Calls out of chaos: the adaptive significance of nonlinear phenomena in mammalian vocal production”. In: *Animal Behaviour* 63.3 (2002), pp. 407–418.

[Nie08] Oriol Nieto. “Voice Transformations for Extreme Vocal Effects”. Master thesis. Pompeu Fabra University, Barelona, Spain, 2008.

[Nis+16] Masanari Nishimura et al. “Singing voice synthesis based on deep neural networks”. In: *INTERSPEECH*. 2016, pp. 2478–2482.

[NLM07] Tin Lay Nwe, Haizhou Li, and Senior Member. “Exploring Vibrato-Motivated Acoustic Features for Singer Identification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.2 (2007), pp. 519–530.

[Nor+08] Karl I Nordstrom et al. “Transforming Perceived Vocal Effort and Breathiness Using Adaptive Pre-Emphasis Linear Prediction”. In: *IEEE transactions on audio, speech, and language processing* 16.6 (2008), pp. 1087–1096.

[Nos+15] Takashi Nose et al. “HMM-based expressive singing voice synthesis with singing style control and robust pitch modeling”. In: *Computer Speech & Language* 34.1 (2015), pp. 308–322.

[NT10] Takayuki Nakata and Sandra E. Trehub. “Expressive timing and dynamics in infant-directed and non-infant-directed singing”. In: *Psychomusicology: Music, Mind & Brain* 21.1 (2010), pp. 130–138.

[Obi11] Nicolas Obin. “MeLos : Analysis and Modelling of Speech Prosody and Speaking Style”. PhD thesis. Université Paris VI - Pierre et Marie Curie (UPMC), 2011, p. 266.

[Ode95] Julian James Odell. “The Use of Context in Large Vocabulary Speech Recognition”. PhD thesis. 1995.

[Ohi+10] Yasunori Ohishi et al. “Statistical Modeling of F0 Dynamics in Singing Voices Based on Gaussian Processes with Multiple Oscillation Bases”. In: *Interspeech*. September. 2010, pp. 2598–2601.

[Ohi+12] Yasunori Ohishi et al. “A Stochastic Model of Singing Voice F0 Contours for Characterizing Expressive Dynamic Components”. In: *13th Annual Conference of the International Speech Communication Association (INTERSPEECH)* 2.1 (2012), pp. 474–477.

[Opp69] Alan V. Oppenheim. “Speech Analysis-Synthesis System Based on Homomorphic Filtering”. In: *The Journal of the Acoustical Society of America* 45.2 (1969), pp. 458–465.

[ORB14] Nicolas Obin, Axel Roebel, and Gregoire Bachman. “On automatic voice casting for expressive speech: Speaker recognition vs. speech classification”. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2014, pp. 950–954.

[Our+10] Keiichiro Oura et al. “Recent development of the HMM-based singing voice synthesis system-Sinsy”. In: *7th ISCA Workshop on Speech Synthesis (SSW-7)*. Kyoto, Japan, 2010, pp. 211–216.

[Our+12] Keiichiro Oura et al. “Pitch adaptive training for HMM-based singing voice synthesis”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE. 2012, pp. 5377–5380.

[OVL12] Nicolas Obin, Christophe Veaux, and Pierre Lanchantin. “Making sense of variations: Introducing alternatives in speech synthesis”. In: *Proceedings of the 6th International Conference on Speech Prosody (SP2012)*. 2012, pp. 179–182.

[Pan+17] Maria Panteli et al. “Towards the characterization of singing styles in world music”. In: *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. 2017, pp. 636–640.

[Par02] Hansang Park. “Time Course of the First Formant Bandwidth”. In: *Annual Meeting of the Berkeley Linguistics Society*. 2002, pp. 213–224.

[PD16] Olivier Perrotin and Christophe D’Alessandro. “Vocal effort modification for singing synthesis”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 08-12-Sept. 2016, pp. 1235–1239.

[Ped11] C. F. Pedersen. “Leja ordering LSFs for accurate estimation of predictor coefficients”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2011, pp. 2545–2548.

[Pee04] Geoffroy Peeters. *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. Tech. rep. 2004.

[Pfl10] Martin Pfleiderer. “Vocal pop pleasures. Theoretical, analytical and empirical approaches to voice and singing in popular music”. In: *IASPM@ Journal* 1.1 (2010), pp. 1–16.

[Por76] M Portnoff. “Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.3 (1976), pp. 243–248.

[PQR99] Michael D. Plumpe, Thomas F. Quatieri, and Douglas A. Reynolds. “Modeling of the glottal flow derivative waveform with application to speaker identification”. In: *IEEE Transactions on Speech and Audio Processing* 7.5 (1999), pp. 569–585.

[Pra94] E. Prame. "Measurements of the vibrato rate of ten singers". In: *The journal of the Acoustical Society of America* 96.4 (1994), pp. 1979–1984.

[PRS08] Yannis Pantazis, Olivier Rosec, and Yannis Stylianou. "On the properties of a time-varying quasi-harmonic model of speech". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2008, pp. 1044–1047.

[PRS11] Yannis Pantazis, Olivier Rosec, and Yannis Stylianou. "Adaptive AM-FM signal decomposition with application to speech analysis". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.2 (2011), pp. 290–300.

[Puc95] Miller Puckette. "Phase-locked vocoder". In: *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*. 1995, pp. 222–225.

[Qui93] J Ross Quinlan. *C4. 5: Programs for machine learning*. San Francisco: Morgan Kaufmann, 1993.

[Rai+11] T Raitio et al. "HMM-Based Speech Synthesis Utilizing Glottal Inverse Filtering". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.1 (2011), pp. 153–165.

[Rec03] ITU Recommendation. "BS.1284-1 General methods for the subjective assessment of sound quality". In: *ITU-R BS* (2003), pp. 1–13.

[RH09] Bernard Roubeau and Nathalie Henrich. "Laryngeal Vibratory Mechanisms: The Notion of Vocal Register Revisited". In: *Journal of voice* 23.4 (2009), pp. 425–438.

[RL08] Dima Ruinskiy and Yizhar Lavner. "Stochastic models of pitch jitter and amplitude shimmer for voice modification". In: *IEEE 25th Convention of Electrical and Electronics Engineers in Israel*. 2008, pp. 489–493.

[RM11] Axel Roebel and S Maller. "Transforming vibrato extent in monophonic sounds". In: *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11)*. Paris, France, 2011.

[Röb08] A Röbel. "Frequency-slope estimation and its application to parameter estimation for non-stationary sinusoids". In: *Computer Music Journal* 32.2 (2008), pp. 68–79.

[Röb10] Axel Röbel. "A Shape-Invariant Phase Vocoder For Speech Transformation". In: *13th International Conference on Digital Audio Effects (DAFx)*. Graz, Austria, 2010.

[Rod02] Xavier Rodet. "Synthesis and processing of the singing voice". In: *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*. Leuven, Belgium, 2002, pp. 99–108.

[Rod09] Xavier Rodet. "Transformation et synthèse de la voix parlée et de la voix chantée". In: *PAROLE ETMUSIQUE*. 2009.

[Roe+12] Axel Roebel et al. "Analysis and modification of excitation source characteristics for singing voice synthesis". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2012, pp. 5381–5384.

[Roe03] Axel Roebel. “A new approach to transient processing in the phase vocoder”. In: *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*. London, UK, 2003.

[RPB84] Xavier Rodet, Yves Potard, and Jean-baptiste Barriere. “The CHANT Project: From the Synthesis of the Singing Voice to Synthesis in General”. In: *Computer Music Journal* 8.3 (1984), pp. 15–31.

[RR05a] Axel Röbel and Xavier Rodet. “Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation”. In: *Proc. of the 8th Int. Conference on Digital Audio Effects (DAFx’05)*. Madrid, Spain, 2005.

[RR05b] Xavier Rodet and Axel Roebel. “Real time signal transposition with envelope preservation in the phase vocoder”. In: *Proc. International Computer Music Conference (ICMC’05)*. 2005, pp. 672–675.

[RVR07] Axel Röbel, Fernando Villavicencio, and Xavier Rodet. “On cepstral and all-pole based spectral envelope modeling with unknown model order”. In: *Pattern Recognition Letters* 28.11 (2007), pp. 1343–1350.

[RW85] S. Roucos and A. Wilgus. “High quality time-scale modification for speech”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP ’85*. 1985, pp. 493–496.

[Sai+04] Takeshi Saitou et al. “Analysis of Acoustic Features Affecting “Singing-ness” and Its Application to Singing-Voice Synthesis from Speaking-Voice”. In: *8th International Conference on Spoken Language Processing - INTERSPEECH*. Jeju Island, Korea, 2004.

[Sai+06] Keijiro Saino et al. “An HMM-based singing voice synthesis system”. In: *9th International Conference on Spoken Language Processing - Interspeech*. February. Pittsburgh, Pennsylvania, 2006, pp. 2274–2277.

[Sai+07] Takeshi Saitou et al. “Speech-to-singing synthesis: converting speaking voices to singing voices by controlling acoustic features unique to singing voices”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY, 2007, pp. 215–218.

[Sak+04] Ken-ichi Sakakibara et al. “Growl Voice in Ethnic and Pop Styles”. In: *Proceedings of the International Symposium on Musical Acoustics*. Nara, Japan, 2004.

[Sak05] Shinsuke Sakai. “Additive modeling of english F0 contour for speech synthesis”. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2005, pp. 277–280.

[San+16] José L. Santacruz et al. “Spectral Envelope Transformation in Singing Voice for Advanced Pitch Shifting”. In: *Applied Sciences* 6.11 (2016), p. 368.

[SDB12] Eric Smialek, Philippe Depalle, and David Brackett. “A spectrographic analysis of vocal techniques in extreme metal for musicological analysis”. In: *ICMC* (2012), pp. 88–93.

[Ser89] Xavier Serra. “A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition”. PhD thesis. Standford University, 1989.

[SF01] Erwin Schoonderwaldt and Anders Friberg. “Toward a rule-based model for violin vibrato”. In: *Workshop on Current Research Directions in Computer Music*. 2001, pp. 61–64.

[SG03] Marc Schroder and Martine Grice. “Expressing vocal effort in concatenative synthesis”. In: *Proc. 15th international conference of phonetic sciences (ICPhS)*. Barcelona, Spain, 2003, pp. 2589–2592.

[SG09] Takeshi Saitou and Masataka Goto. “Acoustic and Perceptual Effects of Vocal training in Amateur Male Singing”. In: *10th Annual Conference of the International Speech Communication Association - Interspeech*. Brighton, UK, 2009, pp. 832–835.

[SG11] Adriana Stan and Mircea Giurgiu. “A Superpositional Model Applied to F0 Parameterization using DCT for Text-to-Speech Synthesis”. In: *6th conference on Speech technology and human-computer dialogue*. 2011.

[Shi+01] Chilin Shih et al. “Prosody Control for Speaking and Singing Styles”. In: *7th European Conference on Speech Communication and Technology - Eurospeech*. Aalborg, Denmark, 2001, pp. 669–672.

[Shi+14] Kanako Shirota et al. “Integration of speaker and pitch adaptive training for HMM-based singing voice synthesis”. In: *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP) INTEGRATION*. 2014, pp. 2578–2582.

[SJ90] Xavier Serra and Julius Smith. “Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition”. In: *Computer Music Journal* 14.4 (1990), p. 12.

[SK03a] Yoshinori Shiga and Simon King. “Estimating the spectral envelope of voiced speech using multi-frame analysis”. In: *Eurospeech*. Geneva, 2003.

[SK03b] Yoshinori Shiga and Simon King. “Estimation of voice source and vocal tract characteristics based on multi-frame analysis”. In: *Eurospeech* 2 (2003), pp. 1749–1752.

[SLG13] Johan Sundberg, Filipa M. B. Lã, and Brian P. Gill. “Formant Tuning Strategies in Professional Male Opera Singers”. In: *Journal of Voice* 27.3 (2013), pp. 278–288.

[SM93] Harm K Schutte and Donald G Miller. “Belting and Pop, Nonclassical Approaches to the Female Middle Voice: Some Preliminary Considerations”. In: *Journal of Voice* 7.2 (1993), pp. 142–150.

[Smi67] Huston Smith. “On an Unusual Mode of Chanting by Certain Tibetan Lamas”. In: *The Journal of the Acoustical Society of America* 41.5 (1967), p. 1262.

[SMK04] Jan P. H. van Santen, Taniya Mishra, and Esther Klabbers. “Estimating Phrase Curves in the General Superpositional Intonation Model”. In: *5th ISCA Speech Synthesis Workshop*. Pittsburgh, PA, USA, 2004, pp. 61–66.

[SO84] Yumi Sasaki and Hiroshi Okamura. “Harmonics-to-noise ratio and psychophysical measurement of the degree of hoarseness”. In: *Journal of Speech and Hearing Research* 27.2-6 (1984).

[Sta11] Ryan Stables. "Towards a model for the humanisation of pitch drift in singing voice synthesis". In: *ICMC*. 2011.

[STK10] Keijiro Saino, Makoto Tachibana, and Hideki Kenmochi. "A Singing Style Modeling System for Singing Voice Synthesizers". In: *Interspeech*. Makuhari, Chiba, Japan, 2010, pp. 2894–2897.

[Sty01] Yannis Stylianou. "Applying the harmonic plus noise model in concatenative speech synthesis". In: *IEEE Transactions on Speech and Audio Processing* 9.1 (2001), pp. 21–29.

[SUA02] Takeshi Saitou, Masashi Unoki, and Masato Akagi. "Extraction of F0 dynamic characteristics and development of F0 control model in singing voice". In: *Proceedings of the 2002 International Conference on Auditory Display*. Kyoto, Japan, 2002.

[SUA05] Takeshi Saitou, Masashi Unoki, and Masato Akagi. "Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis". In: *Speech Communication* 46 (2005), pp. 405–417.

[Sun01] Johan Sundberg. "Level and center frequency of the singer's formant". In: *Journal of Voice* 15.2 (2001), pp. 176–186.

[Sun06] Johan Sundberg. "The KTH synthesis of singing". In: *Advances in Cognitive Psychology* 2.2-3 (2006), pp. 131–143.

[Sun07] Johan Sundberg. "Synthesising Singing". In: *Proceedings SMC'07, 4th Sound andMusic Computing Conference*. July. Lefkada, Greece Synthesising, 2007, pp. 9–13.

[Sun90] Johan Sundberg. *The science of singing voice*. 1990.

[Tam+01a] Masatsune Tamura et al. "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2001, pp. 805–808.

[Tam+01b] Masatsune Tamura et al. "Text-to-speech synthesis with arbitrary speaker's voice from average voice". In: *7th European Conference on Speech Communication and Technology - Eurospeech*. 2001, pp. 345–348.

[TBC98] Paul Taylor, Alan W Black, and Richard Caley. "The Architecture of the Festival Speech Synthesis System". In: *Proc. 3rd ESCA Workshop on Speech Synthesis*. 1998, pp. 147–151.

[TD04] F. Thibault and P. Depalle. "Adaptive processing of singing voice timbre". In: *Proceedings of the Canadian Conference on Electrical and Computer Engineering* 2.1 (2004), pp. 871–874.

[TD12] J Tilmanne and Thierry Dutoit. "Continuous control of style and style transitions through linear interpolation in hidden markov model based walk synthesis". In: *Transactions on Computational Science XVI* (2012), pp. 34–54.

[TE00] Hartmut Traunmüller and Anders Eriksson. "Acoustic effects of variation in vocal effort by men, women, and children". In: *Journal of the Acoustical Society of America* 107.6 (2000), pp. 3438–3451.

[Ter74] E Terhardt. "On the perception of periodic sound fluctuations (roughness)". In: *Acta Acustica united with Acustica* 30.4 (1974), pp. 201–213.

[Tig+97] Monika Tigges et al. “Observation and modelling of glottal biphonation”. In: *Acta Acustica united with Acustica* 83.4 (1997), pp. 707–714.

[TS01] M Thalén and J Sundberg. “Describing different styles of singing: a comparison of a female singer’s voice source in "Classical", "Pop", "Jazz" and "Blues"”. In: *Logopedics, phoniatrics, vocology* 26.2 (2001), pp. 82–93.

[Tsa+10] Author Chen-gia Tsai et al. “Aggressiveness of the Growl-Like Timbre: Acoustic Characteristics, Musical Implications, and Biomechanical Mechanisms”. In: *Music Perception: An Interdisciplinary Journal* 27.3 (2010), pp. 209–222.

[Tur+05] Oytun Turk et al. “Voice Quality Interpolation for Emotional Text-To-Speech Synthesis”. In: *9th European Conference on Speech Communication and Technology*. 2005.

[TW09] Ingo Titze and Albert S Worley. “Modeling source-filter interaction in belting and high-pitched operatic male singing”. In: *Journal of the Acoustical Society of America* 126.3 (2009), p. 1530.

[TWR08] Jonathan Teutenberg, Catherine Watson, and Patricia Riddle. “Modelling and synthesizing F0 contours with the discrete cosine transform”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2008, pp. 3973–3976.

[UBB13a] Marti Umbert, Jordi Bonada, and Merlijn Blaauw. “Generating singing voice expression contours based on unit selection”. In: *Stockholm Music Acoustics Conference (SMAC)*. 2013, pp. 315–320.

[UBB13b] Marti Umbert, Jordi Bonada, and Merlijn Blaauw. “Systematic database creation for expressive singing voice synthesis control”. In: *8th ISCA Workshop on Speech Synthesis*. 2013, pp. 213–216.

[Umb+15] Marti Umbert et al. “Expression Control in Singing Voice Synthesis: Features, approaches, evaluation, and challenges”. In: *IEEE Signal Processing Magazine* 32.6 (2015), pp. 55–73.

[Umb15] Marti Umbert. “Expression Control of Singing Voice Synthesis: Modeling Pitch and Dynamics with Unit Selection and Statistical Approaches”. PhD thesis. Universitat Pompeu Fabra, Barcelona, Spain, 2015.

[Une02] Marcus Uneson. “Burcas - A Simple Concatenation-based MIDI-to-Singing Voice Synthesis System for Swedish”. Master thesis. Lund University, 2002.

[Van+16] Aäron Van Den Oord et al. “Wavenet: a generative model for raw audio”. In: *arXiv* (2016).

[Van58] J Van Den Berg. “Myoelastic-aerodynamic theory of voice production”. In: *Journal of Speech, Language, and Hearing Research* 1.3 (1958), pp. 227–244.

[Vep04] Jithendra Vepa. “Joint cost for unit selection speech synthesis”. PhD thesis. University of Edinburgh, 2004.

[VK05] Ashish Verma and Arun Kumar. “Introducing roughness in individuality transformation through jitter modeling and modification”. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’05)*. 2005, pp. 5–8.

[VMT92] H Valbret, E. Moulines, and J.P. Tubach. “Voice transformation using PSOLA technique”. In: *Speech Communication* 11.2-3 (1992), pp. 175–187.

[VRR06] Fernando Villavicencio, Axel Röbel, and Xavier Rodet. “Improving LPC spectral envelope extraction of voiced speech by true-envelope estimation”. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2006, pp. 869–872.

[VRR07] Fernando Villavicencio, Axel Röbel, and Xavier Rodet. “All-pole spectral envelope modelling with order selection for harmonic signals”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2007.

[Wak73] Hisashi Wakita. “Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms”. In: *IEEE Transactions on Audio and Electroacoustics* 21.5 (1973), pp. 417–427.

[WG04] Gerhard Widmer and Werner Goebl. “Computational Models of Expressive Music Performance: The State of the Art”. In: *Journal of New Music Research* 33.3 (2004), pp. 203–216.

[Wis07] Timothy Wise. “Yodel species: a typology of falsetto effects in popular music vocal styles”. In: *Radical Musicology* 2.2007 (2007), p. 57.

[Yan08] Yannis Stylianou Yannis Pantazis. “Improving the modeling of the noise part in the harmonic plus noise model of speech”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2008, pp. 4609–4612.

[Yos+00] Takayoshi Yoshimura et al. “Speaker interpolation for HMM-based speech synthesis system”. In: *Acoustical Science and Technology* 21.4 (2000), pp. 199–206.

[Yos+01] Takayoshi Yoshimura et al. “Mixed excitation for HMM-based speech synthesis”. In: *7th European Conference on Speech Communication and Technology Eurospeech’01*. 2001, pp. 2263–2266.

[Yos+99] Takayoshi Yoshimura et al. “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis”. In: *6th European Conference on Speech Communication and Technology*. 1999.

[YRR10] Chunghsin Yeh, Axel Roebel, and Xavier Rodet. “Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals”. In: *IEEE Transactions on Audio, Speech and Language Processing* 18.6 (2010), pp. 1116–1126.

[Zöl11] Udo Zölzer. *DAFX : Digital Audio Effects*. Vol. 4. 2011.

[ZTB09] Heiga Zen, Keiichi Tokuda, and Alan W. Black. “Statistical parametric speech synthesis”. In: *Speech Communication* 51.11 (2009), pp. 1039–1064.

[Zwi60] Eberhard Zwicker. “Ein Verfahren zur Beredinung der Lautst{ä}rke”. In: *Acta Acustica united with Acustica* 10.4 (1960), pp. 304–308.

[Zwi61] E. Zwicker. “Subdivision of the Audible Frequency Range into Critical Bands”. In: *The Journal of the Acoustical Society of America* 33.2 (1961), pp. 248–248.